**How to Use Firebug for Creating Selenium Scripts – Selenium Tutorial #4**

comprised of installation of Firebug and its usability.

***lets learn how to use Firebug add-on for creating Selenium scripts. In the process we will also learn how to install Firebug.***

**Introduction to Firebug**

Firebug is a Mozilla Firefox add-on. This tool helps us in identifying or to be more particular inspecting HTML, CSS and JavaScript elements on a web page. It helps us identifying the elements uniquely on a webpage. The elements can be found uniquely based on their locator types which we would be discussing later in this tutorial.

**How to Install Firebug?**

For the ease of understanding, we would bifurcate the installation process into the following steps.

**Step -1:** Launch the Mozilla Firefox browser and navigate to this[Firebug add-on download page](http://addons.mozilla.org/en-US/firefox/addon/firebug/). The URL takes us to Firefox add-ons section.

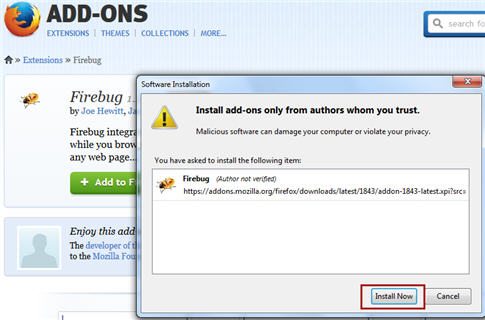
**Step -2:** Click on the “Add to Firefox” button present on the webpage. Refer the following figure for the same.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/install-firebug.jpg)

**Step-3:** As soon as we click on the “Add to Firefox” button, a security alert box would appear, click on the “Allow” button now.

**Step-4:** Now Firefox downloads the add-on in the backdrop and a progress bar is displayed.

**Step-5:** As soon as the process completes, software installation window appears. Now click on the “Install Now” button.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/install-firebug-1.jpg)

**Step-6:** As soon as the installation completes, a pop up appears saying that the firebug has been installed successfully. Now choose to close this pop up.

*Note*: Unlike Selenium IDE, we are not required to restart the Firefox to reflect the firebug installation, rather it comes readily.

**Step-7:** Now to launch firebug, we can opt either of the following ways:

* Press F12
* Click on the firebug icon present in the extreme top-right corner of the Firefox window.
* [install firebug 2](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/install-firebug-2.jpg)
* Click on Firefox menu bar -> Web Developer -> firebug -> Open Firebug.

**Step-8**: Now the firebug can be seen at the bottom of the Firefox window.

Now that we have downloaded and installed firebug, let’s move ahead with the types of locators that we would be creating using firebug.

**Creating Selenium Script using Firebug**

Unlike Selenium IDE, In Firebug, we create automated test scripts manually by adding multiple test steps to form a logical and consistent test script.

Let us follow a progressive approach and understand the process step by step.

**Scenario:**

* Open “https://accounts.google.com”.
* Assert Title of the application
* Enter an invalid username and invalid password and submit the details to login.

**Step 1 –** Launch the Firefox and open Selenium IDE from the menu bar.

**Step 2 –** Enter the address of application under test (“https://accounts.google.com”) inside the Base URL textbox.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-Script-using-Firebug-1.jpg)

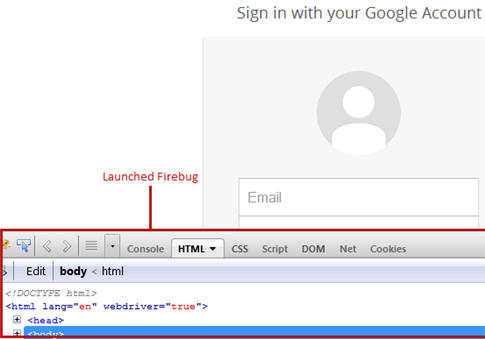
**Step 3 –** By default, the Record button is in ON state. Remember to tune it OFF state so as to disable the recording mode. Notice if the recording mode is in ON state, it may result in recording our interactions with the web browser.

[Selenium Script using Firebug 2](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-Script-using-Firebug-2.jpg)

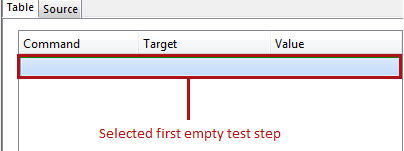
[Selenium Script using Firebug 3](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-Script-using-Firebug-3.jpg)

**Step 4 –** Open the application under test (https://accounts.google.com) in the Firefox.

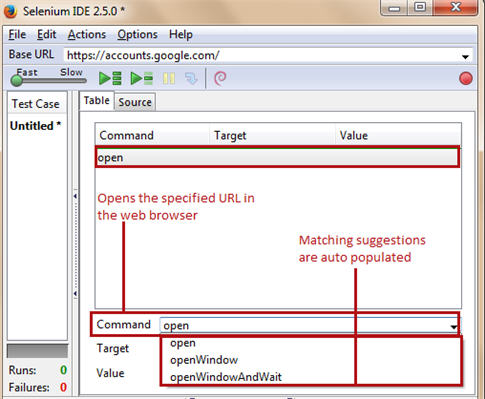
**Step 5 –** Launch Firebug in the web browser.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-Script-using-Firebug-4.jpg)

**Step 6 –** Select the empty test step within the Editor.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-Script-using-Firebug-5.jpg)

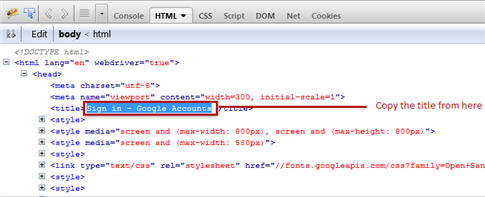
**Step 7 –** Type “open” in the command text box present in the Editor Pane. The “open” command opens the specified URL in the web browser.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-Script-using-Firebug-6.jpg)

Recommendation: While typing commands in the command text box, user can leverage the feature of auto selection. Thus, as soon as the user types a sequence of characters, the matching suggestions would be auto populated.

User can also click on the dropdown available within the command text box to look at all the commands provided by Selenium IDE.

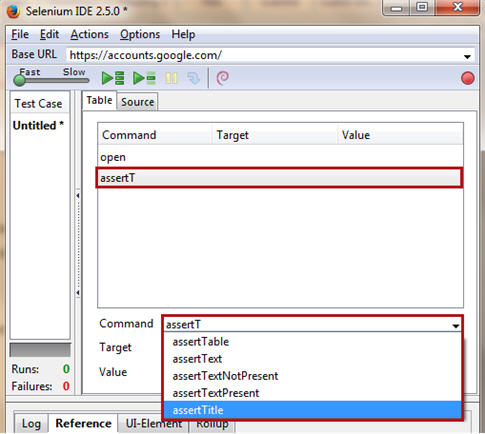
**Step 8 –** Now, motion towards the Firebug section within the web browser, expand “head” section of the HTML code. Notice the HTML tag <title>. Thus to assert the title of the webpage, we would require the value of the <title> tag.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-Script-using-Firebug-7.jpg)

Copy the title of the webpage which is “Sign in – Google Accounts” in our case.

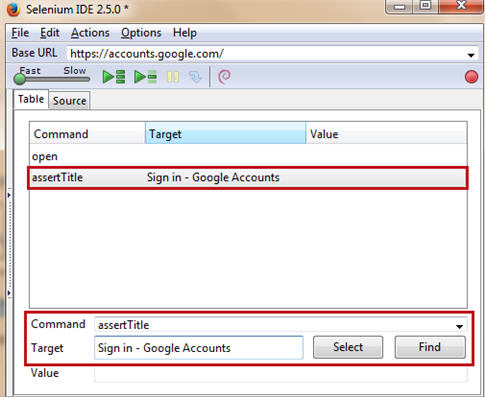
**Step 9 –** Select the second empty test step within the Editor.

**Step 10 –** Type “assertTitle” in the command text box present in the Editor Pane. The “assertTitle” command returns the current page title and compares it with the specified title.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-Script-using-Firebug-8.jpg)

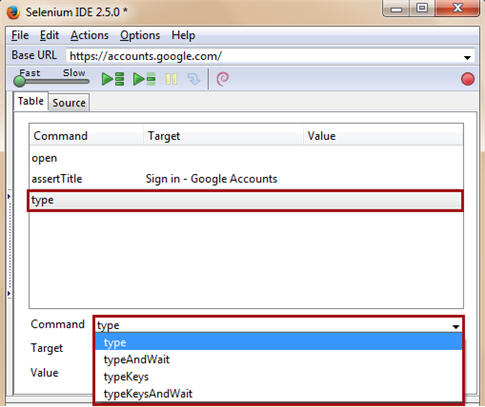
------------

**Step 11 –** Paste the title copied in step 8 into the Target field of the second.

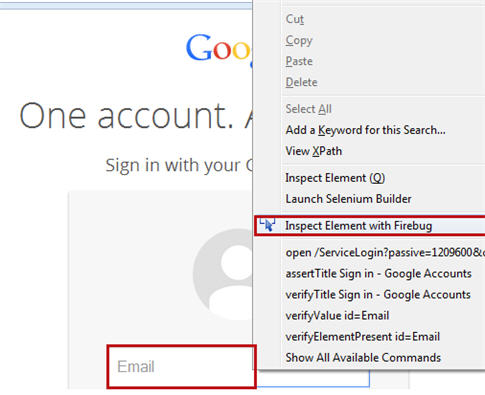
[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-Script-using-Firebug-9.jpg)

**Step 12 –** Now select the third empty test step in the Editor Pane

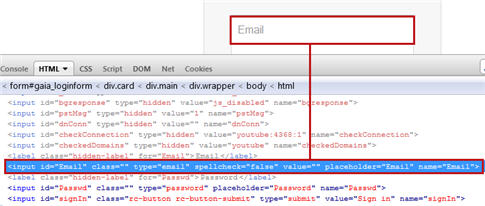
**Step 13 –** Type “type” command within the command text box. The “type” command enters a value in the specified web element on to the GUI.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-Script-using-Firebug-10.jpg)

**Step 14 –** Now switch to the web browser, bring the mouse cursor to the “Email” textbox within the login form and press a right click.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-Script-using-Firebug-11.jpg)

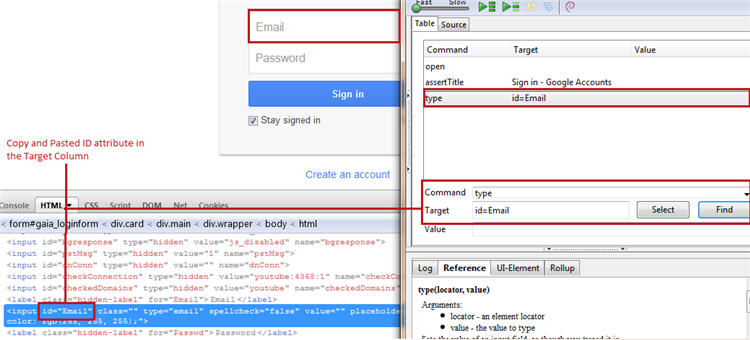
Choose “Inspect Element with Firebug”option. Notice that the Firebug automatically highlights the corresponding HTML code for the web element i.e. “Email Textbox”.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-Script-using-Firebug-12.jpg)

**Step 15 –** The HTML code in the above illustration manifests the distinct property attributes belonging to the “Email” text box. Notice that there are four properties (ID, type, placeholder and name) that uniquely identify the web element on the web page. Thus it’s up to the user to choose one or more than one property to identify the web element.

Thus, in this case, we choose ID as the locator. Copy the ID value and paste it in the Target field of the third test step prefixed with “id=” to indicate Selenium IDE to locate a web element having ID as “Email”.

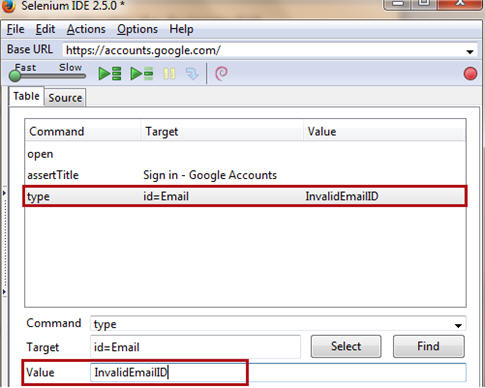
***(Click to view enlarged image)***

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-Script-using-Firebug-13.jpg)

Make a note that Selenium IDE is case sensitive, thus type the attribute value carefully and precisely the same as it is displayed in the HTML code.

**Step 16 –** Click on the Find button to verify if the locator selected finds and locates the designated UI element on the web page.

**Step 17 –** Now, the next step is to enter the test data in the Value textbox of the third test step within the Editor Pane. Enter “InvalidEmailID” in the Value textbox. User can alter the test data as and when it is desired.

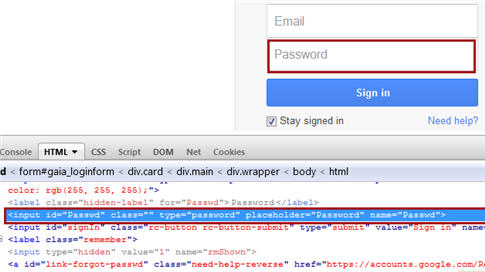
[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-Script-using-Firebug-14.jpg)

**Step 18 –** Now select the fourth empty test step in the Editor Pane

**Step 19 –** Type “type” command within the command text box.

**Step 20 –** Now switch to the web browser, bring the mouse cursor to the “Password” textbox within the login form and press a right click.

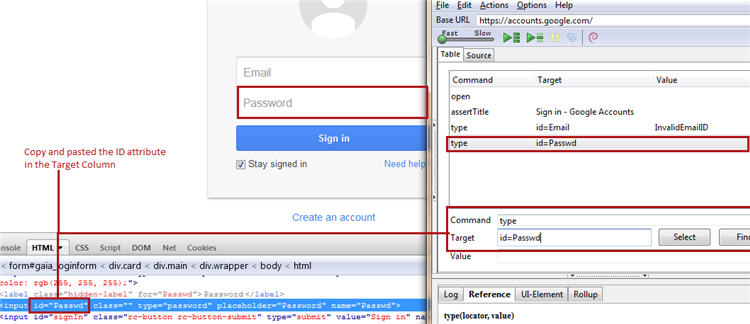
Choose “Inspect Element with Firebug”option.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-Script-using-Firebug-15.jpg)

**Step 21 –** The HTML code below manifests the distinct property attributes belonging to the “Password” text box. Notice that there are four properties (ID, type, placeholder and name) that uniquely identify the web element on the web page. Thus it’s up to the user to choose one or more than one property to identify the web element.

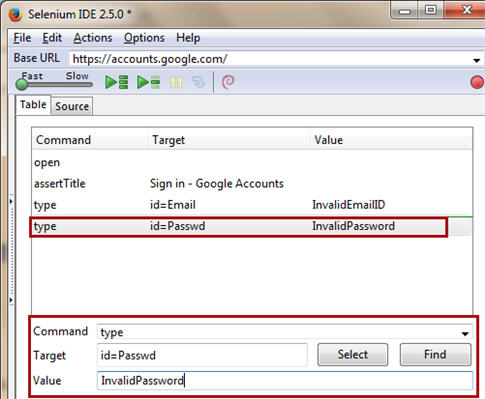
Thus, in this case, we choose ID as the locator. Copy the ID value and paste it in the Target field of the third test step prefixed with “id=”.

***(Click to view enlarged image)***

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-Script-using-Firebug-16.jpg)

**Step 22 –** Click on the Find button to verify if the locator tabbed finds and locates the designated UI element on the web page.

**Step 23 –** Now, the next step is to enter the test data in the Value textbox of the fourth test step within the Editor Pane. Enter “InvalidPassword” in the Value textbox. User can alter the test data as and when it is desired.

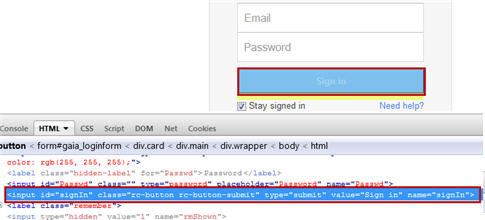
[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-Script-using-Firebug-17.jpg)

**Step 24 –** Now select the fifth empty test step in the Editor Pane

**Step 25 –** Type “click” command within the command text box. The “click” command clicks on a specified web element within the web page.

**Step 26 –** Now switch to the web browser, bring the mouse cursor to the “Sign in” button within the login form and press a right click.

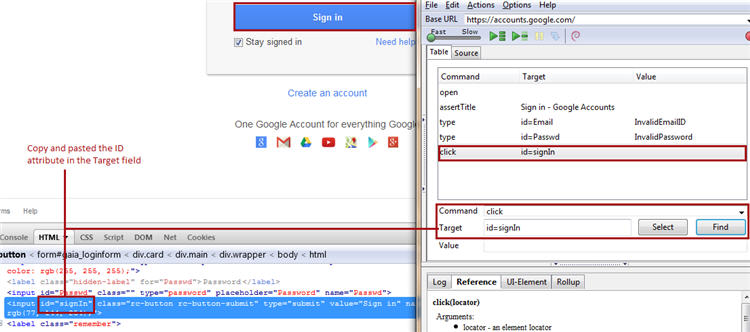
Choose “Inspect Element with Firebug”option.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-Script-using-Firebug-18.jpg)

**Step 27 –** The HTML code below manifests the distinct property attributes belonging to the “Sign in” button.

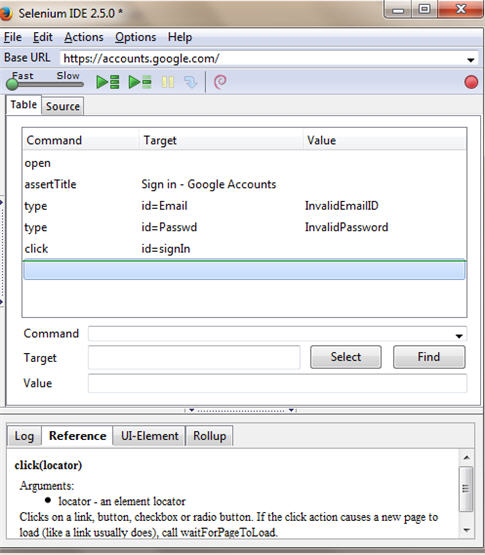
Choose ID as the locator. Copy the ID value and paste it in the Target field of the third test step prefixed with “id=”.

***(Click to view enlarged image)***

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-Script-using-Firebug-19.jpg)

**Step 28 –** Click on the Find button to verify if the locator picked finds and locates the designated UI element on the web page.

The test script is completed now. Refer the following illustration to view the finished test script.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-Script-using-Firebug-20.jpg)

**Step 29 –** Play back the created test script and Save it in the same way we did in the previous tutorial.

**Conclusion**

In this tutorial, we introduced yet another script creation tool or rather a tool that aids to script creation.

Firebug surprisingly has a great potential to locate web elements on a web page. Thus the user can leverage the tool’s capabilities in creating effective and efficient automation test scripts manually.