http://www.softwaretestinghelp.com/firebug-for-selenium-scripts-selenium-tutorial-4/

Tutorial 4

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.

**Install Firebug**

|  |  |
| --- | --- |
| **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/). (<https://addons.mozilla.org/en-US/firefox/addon/firebug/>) The URL takes us to Firefox add-ons section.  **Step -2:** Search Firebug |  |
| **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. |  |

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.
* Click on Firefox menu bar -> Web Developer -> firebug -> Open Firebug.
* [install firebug 2](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/install-firebug-2.jpg)

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

**Creating Selenium Script using Firebug**

At the end of this tutorial we will create manually the following steps.

|  |  |
| --- | --- |
| **If the web requests email** | **If the web doesn’t request email** |
|  |  |

|  |  |
| --- | --- |
| **With Selenium IDE** | **With Firebug** |
| The script is created by recording each step, so we need to enable the record button | The script is created manually, so we need to disable the record button |

|  |  |
| --- | --- |
| **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. |  |
| **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, in another tab. |  |
| **Step 5 –** Launch Firebug in the web browser. Press F-12 |  |
| **Step 6 –** Select the empty test step within the Editor. | [Selenium Script using Firebug 5](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.  **Recommendation:** While typing commands in the command text box, user can click on the dropdown available within the command text box to look at all the commands provided by Selenium IDE. |  |
| **Step 8 –** 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.  Copy the title of the webpage which is “**Sign in – Google Accounts**” |  |
| **Step 9 –** Select the second empty test step within the Editor  Command “assertTitle”  Target “Sign in – Google Accounts”  This is the title copied from Firebug |  |
| **Step 10 –** Switch to the web browser, bring the mouse cursor to the “Email” textbox within the login form and press a right click.  Choose “Inspect Element with Firebug”option.  Notice that the Firebug automatically highlights the corresponding HTML code for the web element i.e. “Email Textbox”. |  |
| **Step 11 –** There are four properties for Email (**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.  In this case, we choose ID as the locator. |  |

**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 12 –** Select the third empty test step within the Editor  Command “type”  Target “id=Email”  Value “InvalidEmailID”  User can alter the test data (value) as and when it is desired.  Click on the Find button to verify if the locator selected finds and locates the designated UI element on the web page. |  |
| **Step 13 –** 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 |  |
| **Step 14 –**There are four properties for Email (**ID, type, placeholder and name**)  Choose id=”Passwd” |  |
| **Step 15 –** Select the fourth empty test step within the Editor  Command “type”  Target “id=Passwd”  Value “InvalidPassword”  User can alter the test data (value) as and when it is desired.  Click on the Find button to verify. | It is id=Passwd not id=”Passwd” |
| **Step 16 –** Interrogate button  Switch to the web browser, bring the mouse cursor to the “Sign In” button within the login form and press a right click. |  |
| **Step 17 –** Select the fifth empty test step within the Editor  Command “click”  Target id=SignIn (not id=”SignIn” )  Click on the Find button to verify. |  |
| The test script is completed now. |  |

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

TO MAKE IT WORK, CHANGE ( InvalidPassword ) for the right password

**Conclusion**

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.

[**Next Tutorial #5**](http://www.softwaretestinghelp.com/using-selenium-xpath-and-other-locators-selenium-tutorial-5/)**:** Moving ahead in the next tutorial, we would have a look at the **various types of locators in Selenium and their accessibility technique to build test scripts**. In the meantime reader can start building his/her automation test scripts using Firebug.