

# Selenium IDE Tutorial For beginners



[testbytes.net/blog/selenium-ide](https://testbytes.net/blog/selenium-ide)

August 7, 2019

[Selenium](#)

Wednesday August 7, 2019

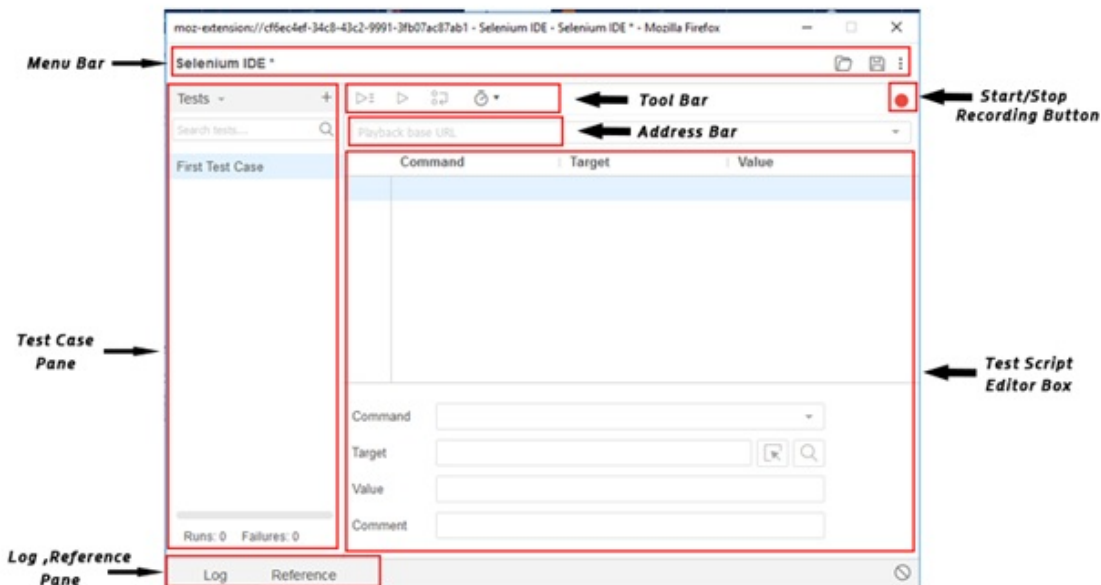


Selenium is the recent craze among the software testers. Every tester is including both automation and manual testers are in a race to learn this tool to stay in the competition. One of the biggest features of Selenium is its IDE. Wish to know what's so special about Selenium IDE?

## About Selenium IDE

*Selenium integrated development environment that is plugged into Firefox. It is an automation testing tool that is very simple, easy and user-friendly. It offers easy installation, learning, and creation of test scripts. Selenium IDE is based on record and playback principle. It is good for all kinds of testing.*

## *Features of Selenium IDE*



Selenium is widely used automation testing tool and offers extensive features. Some of the common features of Selenium IDE are:

- It offers easy and simple record and playback features
- Supports Intelligent field selection
- Auto completion of Selenium commands
- Walk through tests
- Easy Debugging capabilities
- Easy setting of breakpoints
- Save tests as HTML, Ruby, Python, C# scripts, or any other format
- Supports Selenium user-extensions.js
- Supports automatic assertion of title for all pages
- Supports easy customization
- Does not require programming skills

### The drawback of Selenium IDE

Selenium IDE is a Firefox plug-in hence it supports only Firefox and the test scripts Created in Selenium IDE can only be executed in Firefox browser.

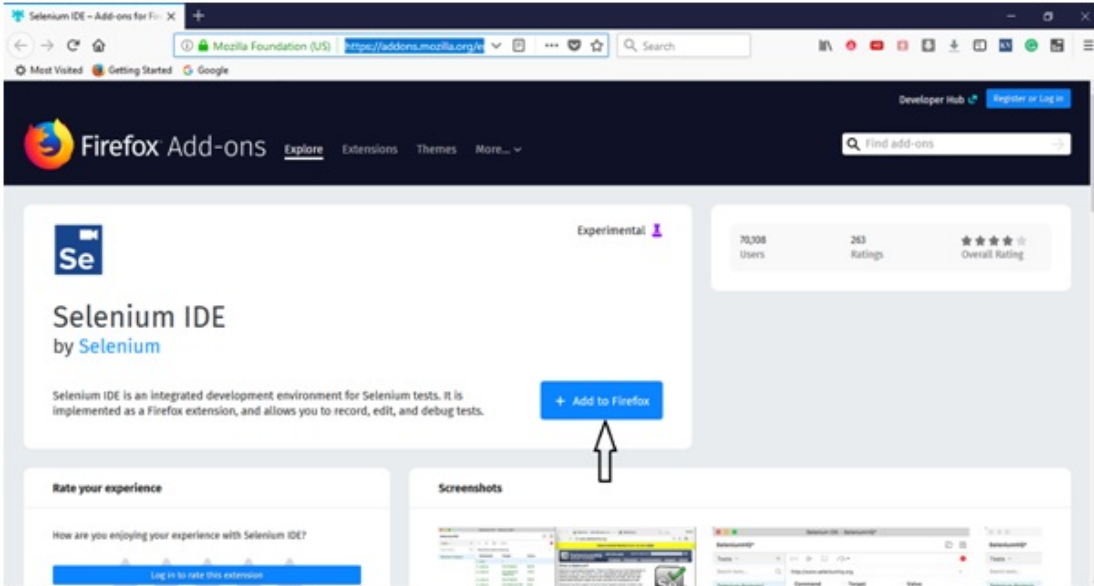
### Downloading and Installing Selenium IDE

Now when we have a good idea on what is Selenium IDE, let us move to the next step of Downloading and Installing Selenium IDE.

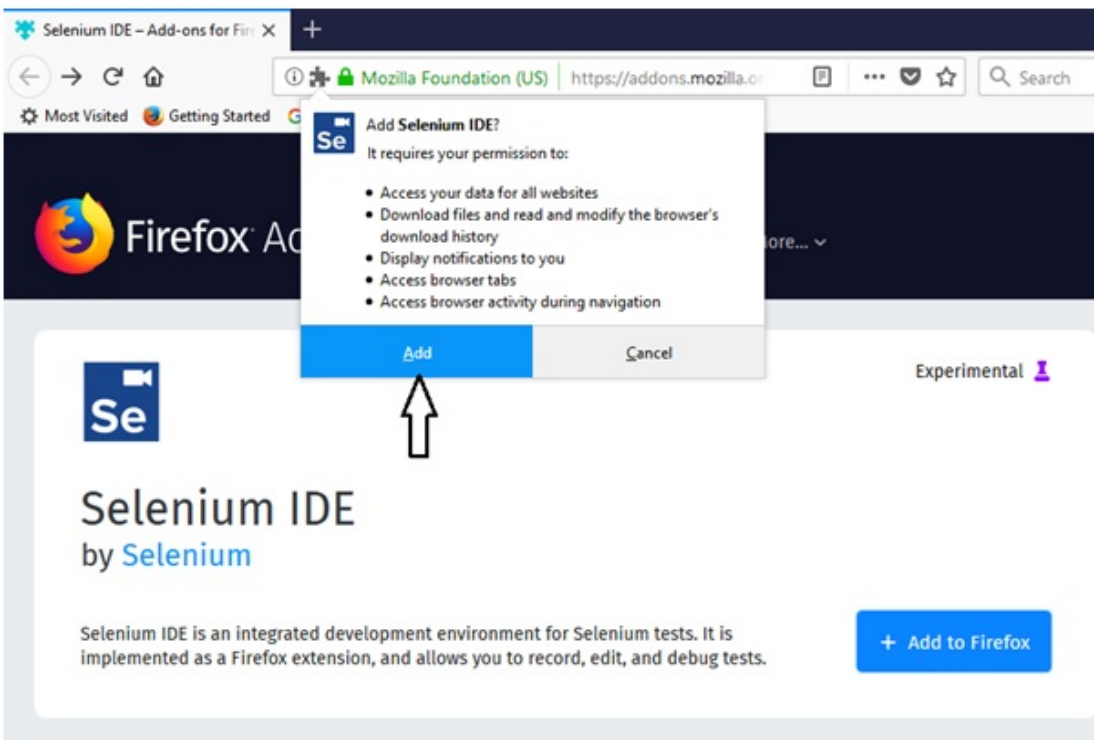
To download Selenium IDE you need to have Mozilla Firefox, if you have it well and good if you don't have it, download it.

### Steps to download and install Selenium IDE

- 1) Launch Mozilla Firefox Browser.
- 2) Open Selenium IDE Add-ons page by typing URL: <https://addons.mozilla.org/en-us/firefox/addon/selenium-ide/> in your browser. Next Click on Add to Firefox button.

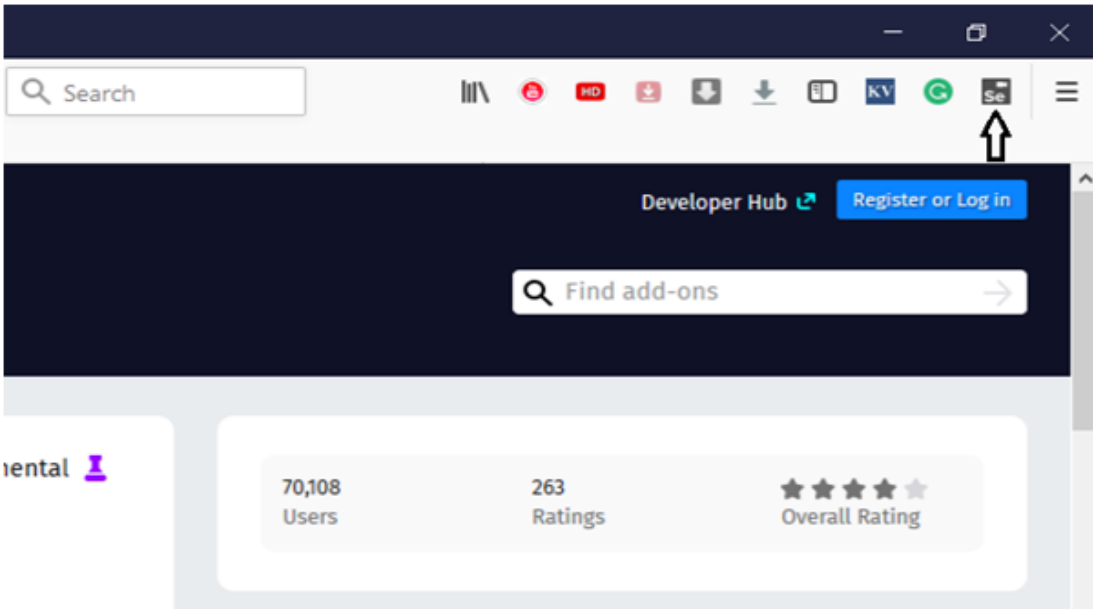


3) You will get a popup asking for your permission to install Selenium IDE Add-ons or not. *Click the Install button over the popup.*



4) Firefox will then install Selenium IDE software and you will get a popup asking you to restart the Firefox. Click the restart button. The Selenium installation will now be reflected on your browser.

5) After you restart your browser you can find the selenium IDE under the tools menu list present at the top bar



6) Click on selenium ide your selenium ide will launch

## Sections of SeleniumIDE

Selenium IDE is divided into different sections. Before start working on it, you must know about these categories:

### Menu Bar

present at the uppermost of the Selenium IDE window. The menu bar consists of five sub-modules.

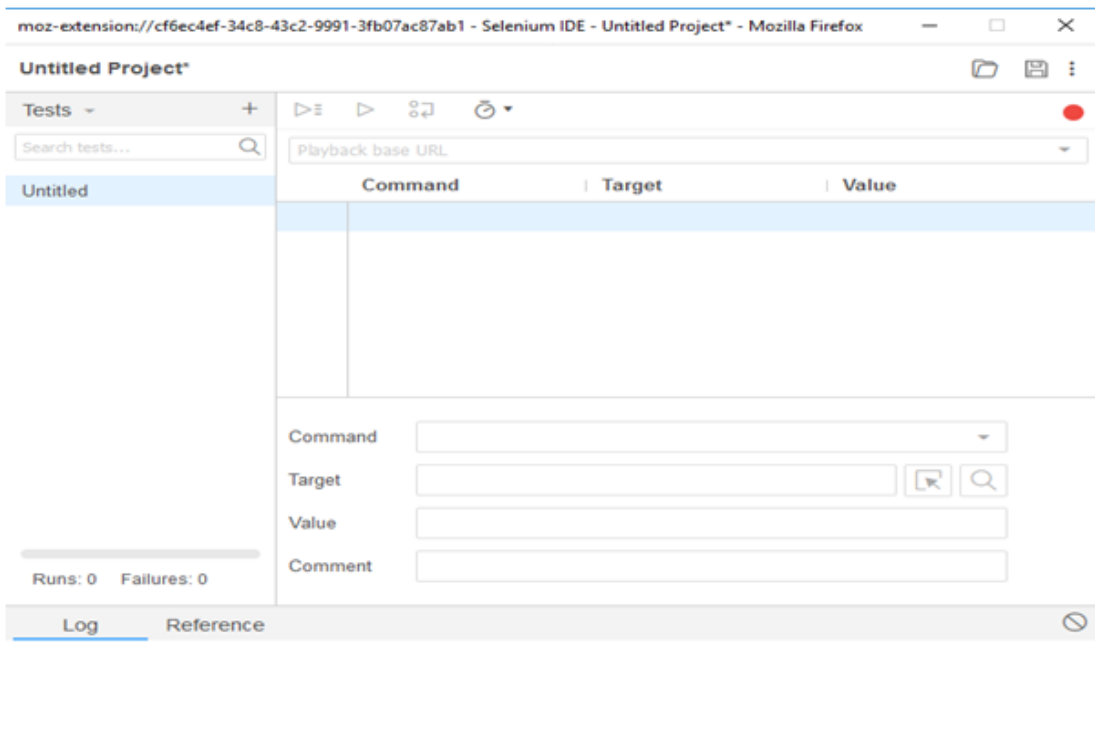
**File Menu:** File Menu *Create, Save and Export Test Case and Test Suite of Selenium IDE.* You can open the file menu by pressing *Alt + F*, or by clicking on *the File* menu.

Under File menu you can find:

- *New Test Case: It creates a new blank Test Case*
- *Open: It Open already saved Test Cases.*
- *Save Test Case: Saves the opened Test case.*
- *Save Test Case As: Saves opened Test Case in a specific location and has a specific name.*
- *Export Test Case As: Assists exporting test cases in various languages like Ruby/Python/Java/C# in both Selenium Remote Control and Selenium WebDriver Format.*
- *Recent Test Cases: Returns a list of few last saved Test Cases.*
- *Add Test Case: Search test cases and merge them to the currently opened test case.*
- *Properties: returns the properties opened test case.*
- *New Test Suite: creates a blank Test Suite*
- *Open Test Suite: Opens existing Test Suite.*
- *Save Test Suite: Saves opened Test Suite.*
- *Save Test Suite As: Saves opened Test Suite in a specific location and has a specific name.*
- *Export Test Suite As: Assists exporting test Suite in various languages like Ruby/Python/Java/C# in both Selenium Remote Control and Selenium WebDriver Format.*
- *Recent Test Suites: Returns a list of few last saved Test Suite.*



## Default Interface of Selenium IDE



**Edit Menu:** *Edit Menu helps to Cut Copy Paste and Insert Command in Selenium IDE Test. You can open the Edit menu by pressing  $Alt + E$ , or by clicking on the Edit menu.*

Under Edit Menu you can find:

- *Undo: last action or few last performed actions are undone*
- *Redo: Re-do the last undone action or series of last undone actions.*
- *Cut: Copies and removes the selected command or test step from current location to some other location in the test.*
- *Copy: Copies selected command or test step from current location to some other location in the test. But it does not remove the command from the previous locations.*
- *Paste: Pastes cut/copied command to a selected location in the test.*
- *Delete: Deletes the chosen command.*
- *Select All: Select all the commands in Test Step Pane.*
- *Insert New Command: Insert a row at a selected location to write new commands*
- *Insert New Comment: Insert a row at a selected location to write a new comment*

**Read also :** [Selenium 4: New Features and Updates](#)

**Actions Menu:** *Helps us to learn Options to Record Play Run And Debug Selenium IDE Test.*

- *Record: It Records the user actions on the webpage via FireFox Web Browser. While recording, the menu item keeps displaying a check against items.*
- *Play Entire test suite: it play-back/re-run every Test Case in the Test Case Pane, following top to bottom approach.*
- *Play current test case: play-back the selected Test Case.*
- *Play test Suites periodically: let the specified test suite to execute at specific times. It is the best way when test suits are required to be rerun periodically as it does not require any human intervention, once specified it will rerun test cases automatically*
- *Toggle Breakpoint:allows you to specify one or more breakpoint(s) to forcefully break the execution at specified steps.*
- *Set / Clear Start Point:it permits the testers to select a start point to start executing the test. It is important for subsequent runs.*
- *Pause / Resume: this enables the pausing and resuming of the test at any point between the test execution.*
- *Step: it permits to step through the playing-back of the test case. It is important for debugging purposes.*
- *Execute this command: This allows testers to execute a particular command instead of executing the complete test case. It is important when testers want to see the behavior of any specific command*
- *Fastest/Faster/Slower/Slowest: it allows to set the execution speed from fastest to lowest concerning the*

*responsiveness of the application under test.*

Other categories available at Menu Bar are:

## Options Menu and Help Menu

### Tool Bar

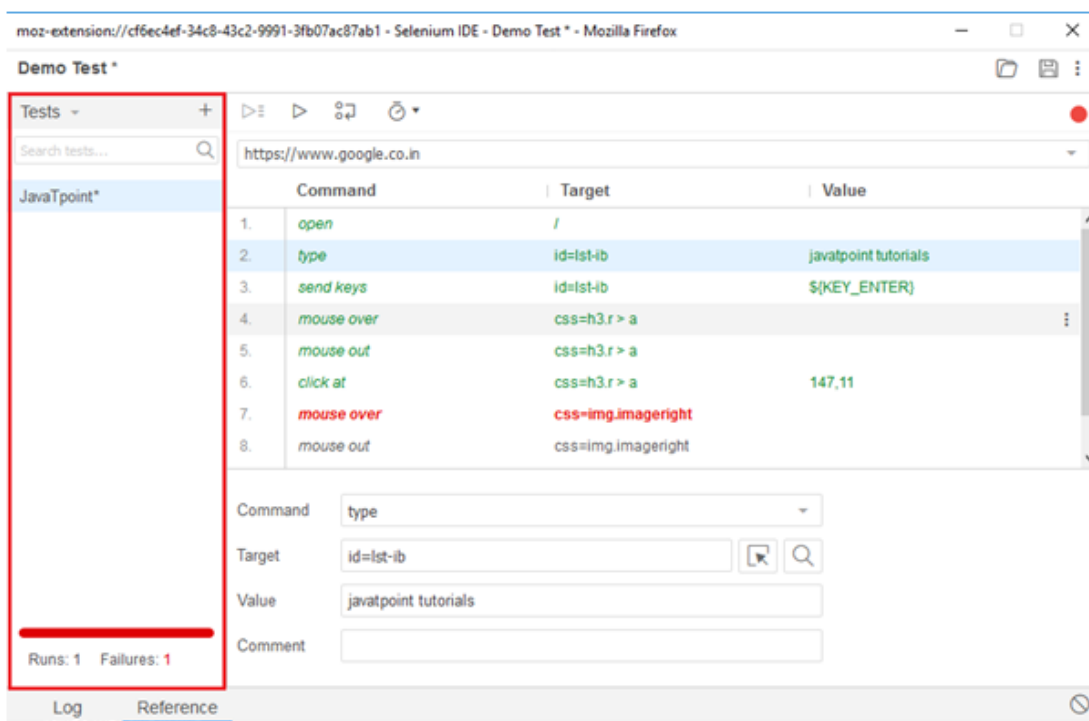
The ToolBar includes *the Buttons that Control the execution* of test cases, for debugging the test cases, setting up the speed of the test, Stopping, Playing and Recording of test cases.

### Test Case Pane

*All the test cases* recorded by IDE are available in Test Case Pane. It can open more than one test case at the same time and supports easy shuffling between the test cases.

It also offers *Test Execution Result Summary* including entire Test Suite status, Total number of Test Executed, etc.

### Test Case Pane



It is the place where [Test Case Steps](#) are recorded. All the user actions are recorded in the order they are performed. It also allows the editing and changing of the test cases

### Output Pane

The bottom pane or the [Log Pane](#) offers the following functions

- Log,
- Reference,
- UI-Element, and
- Rollup

The function performed depend upon which tab is selected.

### Record the Selenium IDE Test Case.

1. Launch Firefox Selenium IDE. Type the URL. Click *the Record* button, a red button on the top right corner. It will record Test Cases.
2. In Firefox, type the same URL as in step 1. Firefox will take you the related webpage.

3. Right click anywhere on this page, you will get Selenium IDE context menu. In the context menu go to *Show Available Commands> Web Page Name*.
4. Click on MyAccount
5. Enter UserName and Password and then click on log in button.
6. To stop the recording click the record button again. Click on *the Table* tab and you will be able to see the recorded commands.
7. Click on the *Source* tab to see *HTML Code*.

**Read also :** [11 Awesome Selenium Alternatives For Testers in 2019](#)

### Save the Selenium IDE Test Case.

1. To save a test case go to File Menu and click on *Save Test Case As*.

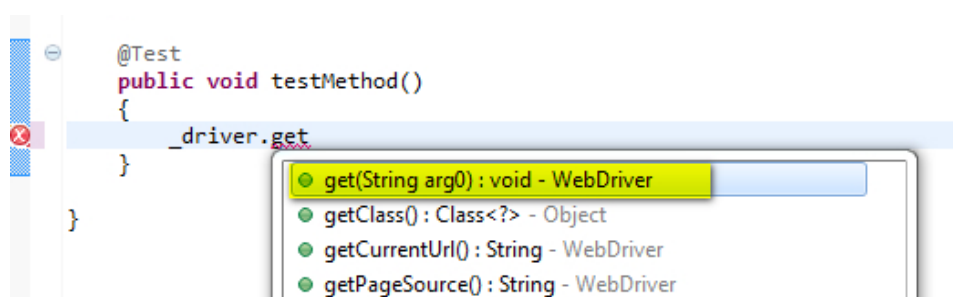
*File -> Save Test Case As.*

2. Choose the desired location and give your file a name and click on Save.
3. You can see the name of the saved test case on the left-hand side.
4. The file will be saved as HTML.

### PlayBack Selenium IDE Test Script

Open a new tab Firefox. Click *the Play* button in Selenium IDE. It will execute the previously recorded tests.

### Selenium Commands



Selenium commands or *Selenese* are a set of test cases that are deployed to test web applications using Selenium.

Selenium commands are of three types:

#### Actions

*Actions* commands control the state of the application. Operations under action commands are:

1. *type this box,*
2. *click this link*
3. *select option.*

Some of these commands can be suffixed with *AndWait* like *clickAndWait*, *typeAndWait*. This prompts Selenium to wait until the web page is loaded. If these commands fail, the existing test is stopped.

#### Accessors

These commands automatically create Assertions and inspect the state of the application.

#### Assertions

They inspect the state of the application adapts to what is anticipated.

They can be further divided into three categories:

- *Assert:*
- *Verify:*
- *WaitFor:*

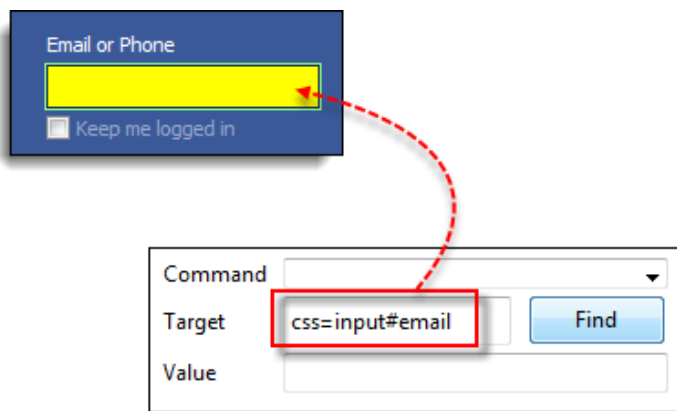
Some of the Commonly used commands in Selenium IDE:

- *type*: Set the value of the input field
- *open*: Opens a web page through the given URL.
- *click*: clicks on checkbox, link, button or radio button.
- *clickAndWait*: When you click on the checkbox, link, button or radio button, and a new page is loaded it calls *waitForPageToLoad*.
- *select*: It select an option drop-down.
- *selectFrame*: It is used to select a frame from the present window.
- *verify title/assert title*: Authenticates an anticipated page title.
- *verifyElementPresent*: Confirms if the indicated element is present on the page.
- *highlight*: Alters the background color of the indicated element.
- *pause*: Wait for the indicated time period
- *echo*: Prints indicated message in your Selenium command tables.

## What are Locators?

Locators in Selenium IDE are used to find and match the elements in the web page that are required to communicate with. The use of right locator promises faster, more trustworthy and low maintenance tests. But choosing the right locators can sometimes become a challenging task.

### Locators in selenium IDE



Locators tell on which GUI elements do Selenium IDE needs to operate on. The correct identification of the locators is very important and it is equally challenging.

There are many commands in Selenium IDE that does not require Locators, but most the commands do require locators. The locators to be used depends on the AUT.

The various types of locator are:

### **Locator: ID**

It is the common way to locate different elements as every element has a unique ID.

**Target Format:** id=*id of the element*

Consider any test app, let it be Facebook.

- Check the "Email" text box using Firebug and note down Locator: ID
- Launch Selenium IDE and write "id=(Locator ID you retrieved in the first step)" in the Target box. When you click on the Find button next to the target box, the "Email ID" text box will be emphasized with a yellow and green border. This means that Selenium IDE has located the "Email ID" text box correctly.

### **Locator: Name**

Locator name is quite similar to Locator: ID. The only difference is that here we use name instead of ID.



**Target Format:** name=*name of the element*.

Consider any test app, let it be Facebook.

- Check the “Email” text box using Firebug and note down Locator: Name
- Launch Selenium IDE and write “Name=(Locator name you retrieved in the first step)” in the Target box. When you click on the Find button next to the target box, the “Email ID” text box will be emphasized with a yellow and green border. This means that Selenium IDE has located the “Email ID” text box correctly.

### **Locator: Link**

This locator is used for hyperlink texts. It can be used by beginning the target with “link=” tailed by hyperlink text.

**Target Format:** link=*link\_text*

Consider any web app as a text app.

- Check any element that has a hyperlink text and using Firebug and notes down Link Text
- Launch Selenium IDE and write “Link =(Link Text you retrieved in the first step)” in the Target box. When you click on the Find button next to the target box, the corresponding text box will be emphasized with a yellow and green border. This means that Selenium IDE has located the element correctly.



### **Locator: CSS**

CSS Selectors are though a complex method to locate elements on a web page, but they are the most preferred method of locating elements in advanced Selenium as they can even detect elements that have no name or no ID.

CSS Selectors are also string patterns That has the ability to recognize an element based on its arrangement of HTML tag, id, class, and attributes. CSS Selectors have many formats, but the most common are:

- Tag and ID
- Tag and class
- Tag and attribute
- Tag, class, and attribute
- Inner text

### **Locator: Xpath**

XPath is a language for navigating DOM (document object model) of a web page. It can locate any element on the web page and is hence the most potent and supple locator.

Some of the Firefox Add-ons that can help in finding XPath of an element:

### **Locator: DOM**

The Document Object Model is an HTML document that can be accessed using JavaScript. It uses hierarchical dotted notation to locate an element on the page.

## Conclusion

If you wish to learn more about selenium we have comprised a tutorial just for you which will take you deep into the tool. [Click here](#)

Method	Target Syntax
By ID	id= id_of_the_element
By Name	name=name_of_the_element
By Name Using Filters	name=name_of_the_element filter=value_of_filter
By Link Text	link=link_text
Tag and ID	<u>css</u> =tag#id
Tag and Class	<u>css</u> =tag.class
Tag and Attribute	<u>css</u> =tag[attribute=value]
Tag, Class, and Attribute	<u>css</u> =tag.class[attribute=value]