**Lab Exercise 9 - Selenium IDE – Assertions Commands**

All commands in Selenium IDE are called the Selenese. With the help of the Selenese commands, all actions can be performed on the web application. Selenese commands are of multiple variants −

* **Actions** − The action commands are used to change the state of the application. For example, close, click, type, open, and so on.
* **Accessors** − The accessor commands are used to gauge the state of the application and store it in variables. For example, store, store title, store text, store value, and so on.
* **Assertions** − The assertion commands are used to verify the state of the application. It has sub-types like the assert, verify, and waitfor.

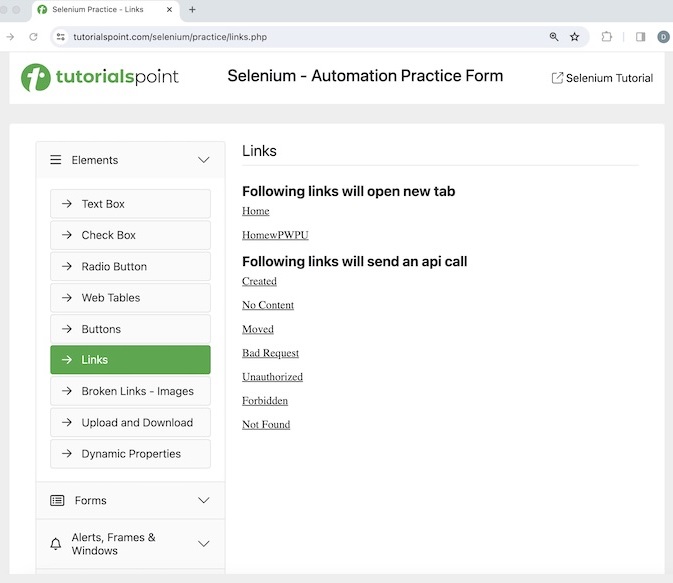
Basic Assertions Commands

The various types of assertions commands are listed below −

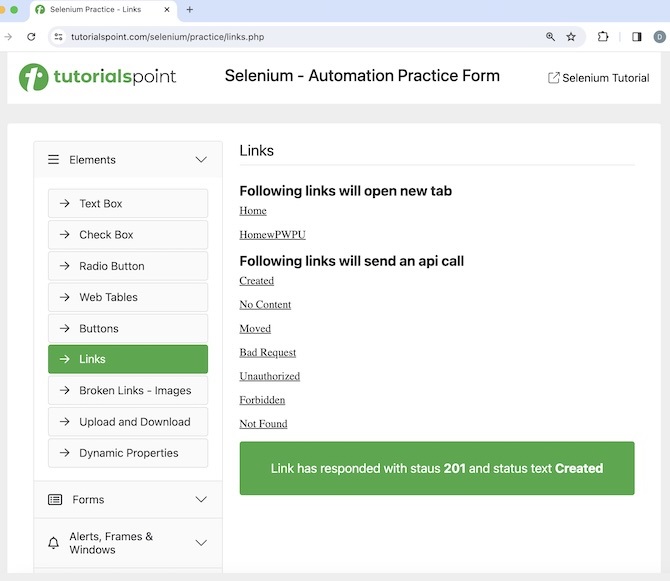
* **assert(variable, expected value)** − This command is used to verify if the variable is matching the expected value. The variable is converted to string for comparison. If assert did not pass, then the execution will stop at the point of mismatch.
* **assert alert(alert text)** − This command is used to verify if an alert has the text matching the alert text. If assert did not pass, then the execution will stop at the point of mismatch.
* **assert checked(locator)** − This command is used to verify if the target element with the locator value is checked. If assert did not pass, then the execution will stop at the point of mismatch.
* **assert confirmation(text)** − This command is used to verify if a confirmation has been rendered. If assert did not pass, then the execution will stop at the point of mismatch.
* **assert editable(locator)** − This command is used to verify if the target element with the locator value is editable. If assert did not pass, then the execution will stop at the point of mismatch.
* **assert element present(locator)** − This command is used to verify if the target element with the locator value is available on the web page. If assert did not pass, then the execution will stop at the point of mismatch.
* **assert element not present(locator)** − This command is used to verify if the target element with the locator value is not available on the web page. If assert did not pass, then the execution will stop at the point of mismatch.
* **assert not checked(locator)** − This command is used to verify if the target element with the locator value is not checked. If assert did not pass, then the execution will stop at the point of mismatch.
* **assert not editable(locator)** − This command is used to verify if the target element with the locator value is not editable. If assert did not pass, then the execution will stop at the point of mismatch.
* **assert not selected value(locator, text)** − .This command is used to verify if the value attribute of the selected option of the target dropdown element does not have the given text. If assert did not pass, then the execution will stop at the point of mismatch.
* **assert not text(locator, text)** − This command is used to verify if the target element with the locator value does not have the given text.
* **assert prompt(text)** − This command is used to verify if a prompt has been rendered. If assert did not pass, then the execution will stop at the point of mismatch.
* **assert selected value(locator, text)** − This command is used to verify if the value attribute of the selected option of the target dropdown element has the given text. If assert did not pass, then the execution will stop at the point of mismatch.
* **assert selected label(locator, text)** − This command is used to verify if the label of the selected option of the target dropdown element has the given text. If assert did not pass, then the execution will stop at the point of mismatch.
* **assert text(locator, text)** − This command is used to verify if the target element with the locator value has the given text. If assert did not pass, then the execution will stop at the point of mismatch.
* **assert value(locator, text)** − This command is used to verify the value of an edit box. For a radio button/checkbox, the value will be set to on/off provided it is selected or not. If assert did not pass, then the execution will stop at the point of mismatch.
* **verify(variable, expected value)** − This command is used to verify if the variable is matching the expected value. The variable is converted to string for comparison. It is a soft assert and execution will not stop in case of a mismatch.
* **verify checked(locator)** − This command is used to verify if the target element with the locator value is checked. It is a soft assert and execution will not stop in case of a mismatch.
* **verify editable(locator)** − This command is used to verify if the target element with the locator value is editable. It is a soft assert and execution will not stop in case of a mismatch.
* **verify element present(locator)** − This command is used to verify if the target element with the locator value is available on the web page. It is a soft assert and execution will not stop in case of a mismatch.
* **verify element not present(locator)** − This command is used to verify if the target element with the locator value is not available on the web page. It is a soft assert and execution will not stop in case of a mismatch.
* **verify not editable(locator)** − This command is used to verify if the target element with the locator value is not editable. It is a soft assert and execution will not stop in case of a mismatch.
* **verify not selected value(locator, text)** − This command is used to verify if the value attribute of the selected option of the target dropdown element does not have the given text. It is a soft assert and execution will not stop in case of a mismatch.
* **verify not text(locator, text)** − This command is used to verify if the target element with the locator value does not have the given text. It is a soft assert and execution will not stop in case of a mismatch.
* **verify selected label(locator, text)** − This command is used to verify if the label of the selected option of the target dropdown element has the given text. It is a soft assert and execution will not stop in case of a mismatch.
* **verify selected value(locator, text)** − This command is used to verify if the value attribute of the selected option of the target dropdown element has the given text. It is a soft assert and execution will not stop in case of a mismatch.
* **verify text(locator, text)** − This command is used to verify if the target element with the locator value has the given text. It is a soft assert and execution will not stop in case of a mismatch.
* **assert title(text)** − This command is used to verify if the alert generated has the given alert text. If assert did not pass, then the execution will stop at the point of mismatch.
* **verify value(locator, text)** − This command is used to verify the value of an edit box. For a radio button/checkbox, the value will be set to on/off provided it is selected or not. It is a soft assert and execution will not stop in case of a mismatch.
* **wait for element editable(locator, wait time)** − This command is used to wait for the given wait time(in milliseconds) for the target element with the locator value to be editable.
* **wait for element not editable(locator, wait time)** − This command is used to wait for the given wait time(in milliseconds) for the target element with the locator value not to be editable.
* **wait for element not present(locator, wait time)** − This command is used to wait for the given wait time(in milliseconds) for the target element with the locator value to be unavailable on the web page.
* **wait for element not visible(locator, wait time)** − This command is used to wait for the given wait time(in milliseconds) for the target element with the locator value to be invisible on the web page.
* **wait for element present(locator, wait time)** − This command is used to wait for the given wait time(in milliseconds) for the target element with the locator value to be available on the web page.
* **wait for element visible(locator, wait time)** − This command is used to wait for the given wait time(in milliseconds) for the target element with the locator value to be visible on the web page.

**Hands-On**

Let us take an example of the below page, where we would click on the **Created** link.



We would verify its page title which is **Selenium Practice - Links**. After clicking the Created link, we would also verify the text - **Link has responded with status 201 and status text Created**.



The steps to be followed are listed below −

**Step 1** − Perform the Steps 1 to 4 as described in the link **Selenium IDE Store Variables**

**Step 2** − Enter open in the Command field, and [**Selenium Automation Practice Form**](https://www.tutorialspoint.com/selenium/practice/links.php) in the Target field, to launch the application.

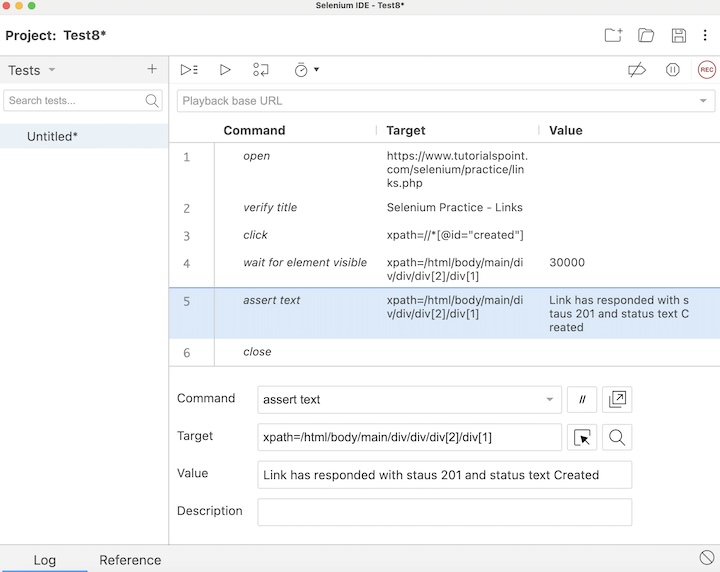
**Step 3** − Enter verify title in the Command field and Selenium Practice - Links in the Target field.

**Step 4** − Enter click in the Command field, xpath=//\*[@id="created"] in the Target field. Please note, xpath=//\*[@id="created"] is the locator value of the link.

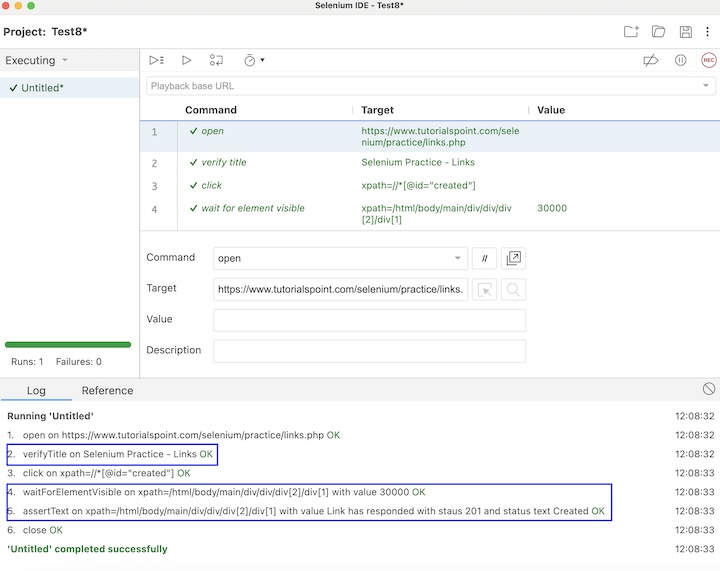
**Step 5** − Enter wait for element visible in the Command field, xpath=/html/body/main/div/div/div[2]/div[1] in the Target field and 30000 in the Value field. Please note, xpath=/html/body/main/div/div/div[2]/div[1] is the locator value of the text to be visible.

**Step 6** − Enter assert text in the Command field, xpath=/html/body/main/div/div/div[2]/div[1] in the Target field and Link has responded with status 201 and status text Created in the Value field. Please note, xpath=/html/body/main/div/div/div[2]/div[1] is the locator value of the text.

**Step 7** − Enter close in the Command field to close browser.



**Step 8** − Click on Run all tests from the top, and hold back for the test execution to end.



In the example above, we had launched the application and verified its page title **Selenium Practice - Links** using the assertions command **verify title**. Then clicked on the **Created** link, and waited for text **Link has responded with status 201 and status text Created** to be visible with the assertion command **wait for element visible** and verified the text with another assertion command assert text.

**Conclusion**

This concludes our comprehensive take on the tutorial on Selenium IDE Assertion Commands. Weve started with describing various assertions commands and an example to walk through how to use the assertions commands along with Selenium. This equips you with in-depth knowledge of the assertions commands in Selenium IDE. It is wise to keep practicing what youve learned and exploring others relevant to Selenium to deepen your understanding and expand your horizons.