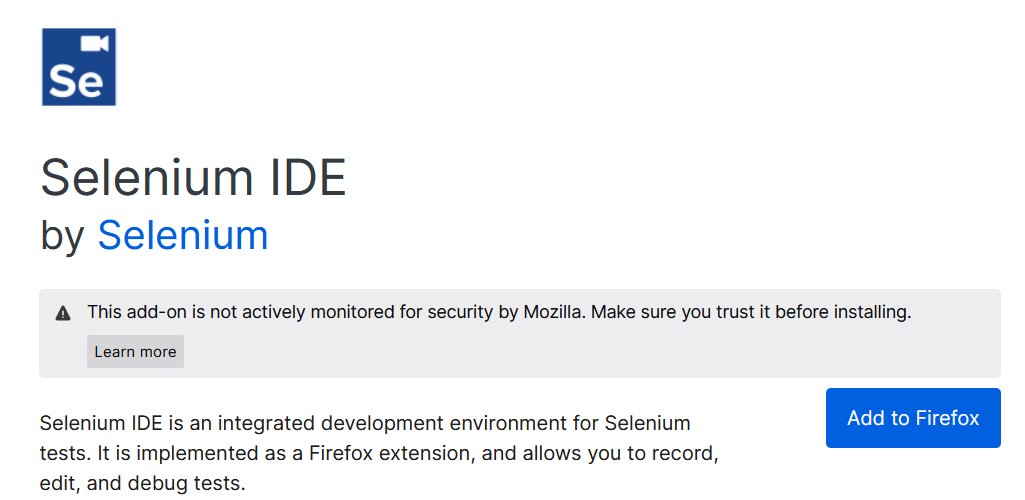
**Selenium IDE**

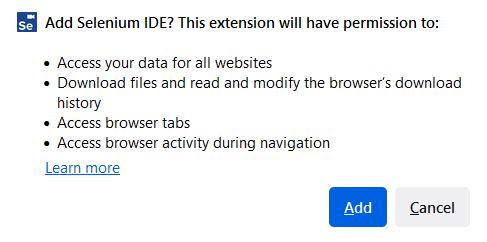
# **Installation in Firefox (Recommended)**

To install Selenium IDE on Firefox, follow the steps below:

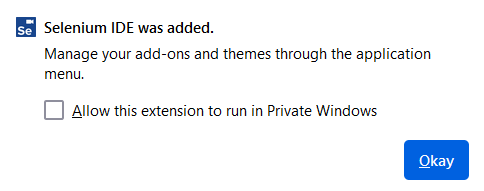
**Step 1:** Open the Firefox browser on your system and then go to the [**Selenium IDE link**](https://addons.mozilla.org/en-US/firefox/addon/selenium-ide/?utm_source=addons.mozilla.org&utm_medium=referral&utm_content=search). This will direct you to the Firefox Extensions. Click on **Add to Firefox** button.



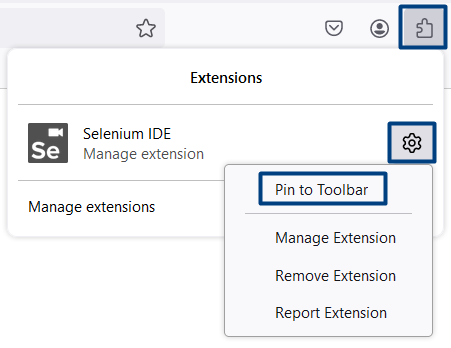
**Step 2:** As you click on Add to Firefox button, a dialog box of warning will appear on the screen. Simply click on the Add button to add the extension to your Firefox browser.



**Step 3 (Optional):**



**Step 4:** Now you need to click on the extension icon on the Firefox browser which is at the top right corner. A list of extensions will appear on the screen, pin the Selenium IDE extension.

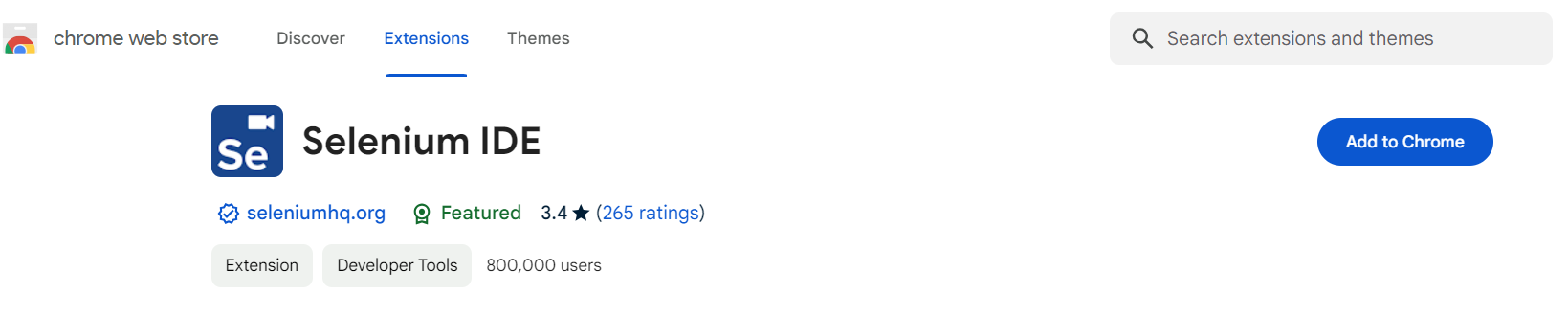


**Step 5:** Now click on Selenium IDE icon to start working with it.

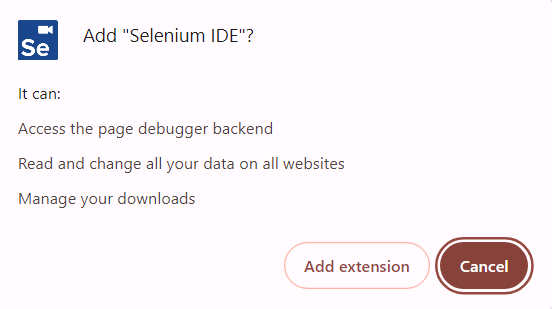
# **Installation in Google Chrome**

To install Selenium IDE on Google Chrome, follow the steps below:

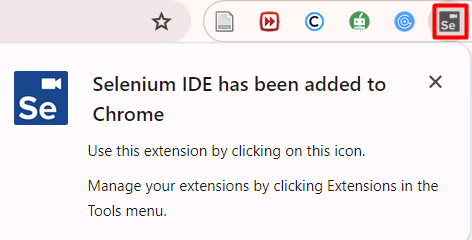
**Step 1:** Open the Google Chrome browser on your system and then go to the [**Selenium IDE link**](https://chromewebstore.google.com/detail/selenium-ide/mooikfkahbdckldjjndioackbalphokd). This will direct you to the Chrome web store. Click on **Add to Chrome** button.



**Step 2:** As you click on Add to Chrome button, a dialog box of warning will appear on the screen. Simply click on the Add extension button to finally add the extension to your Chrome browser.



**Step 3:** Now you need to click on the extension icon on the Google Chrome browser which is at the top right corner. A list of extensions will appear on the screen, pin the Selenium IDE extension.

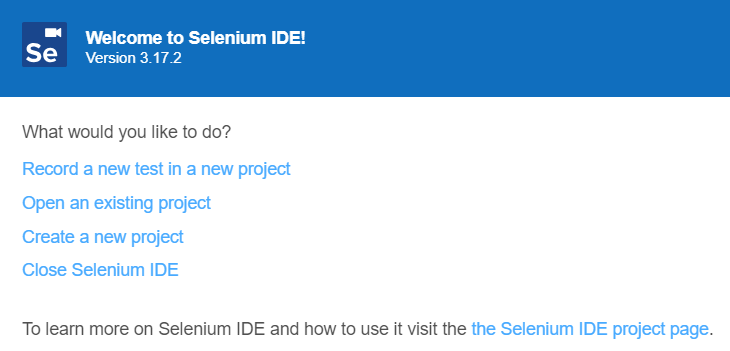


Pin Selenium IDE

**Step 4:** Now click on Selenium IDE to start working with it.

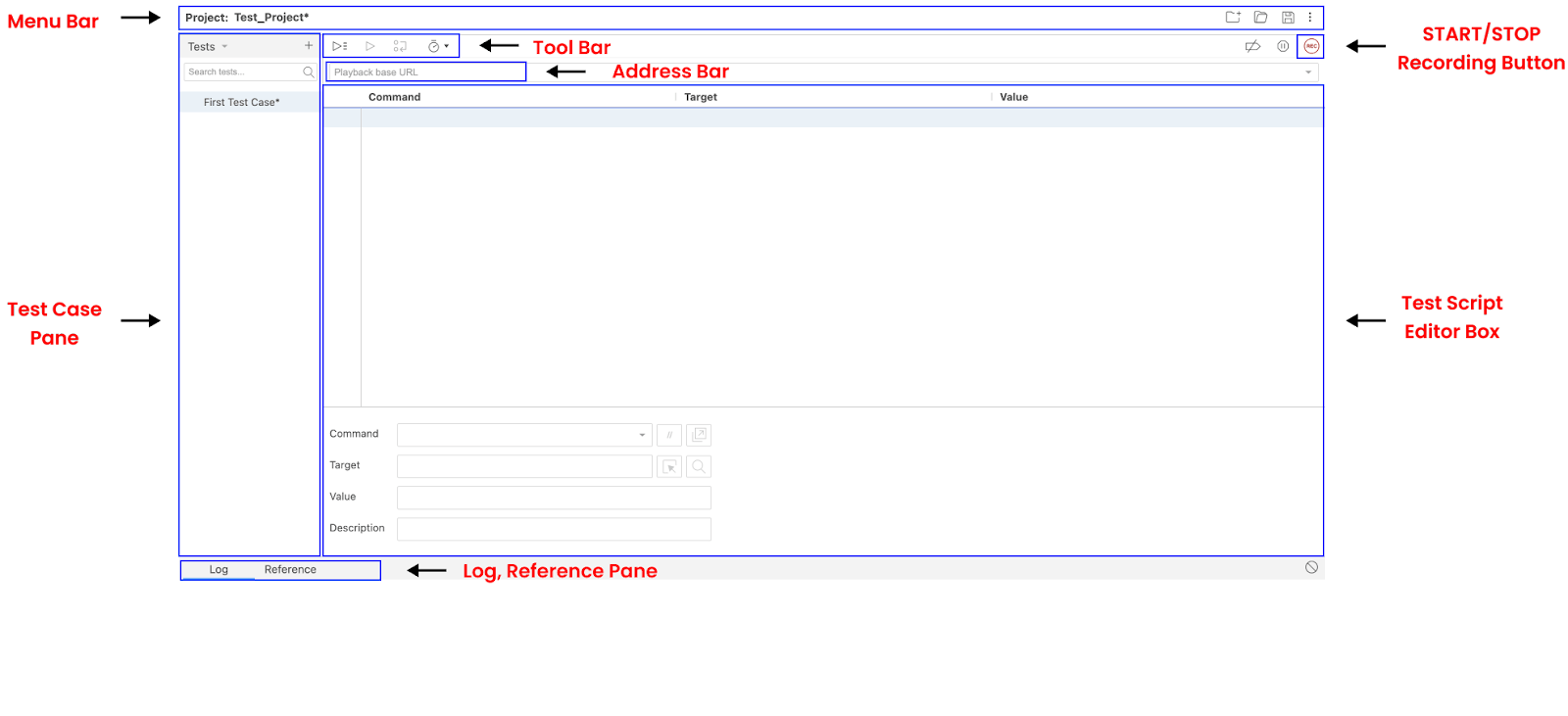
# **Welcome and Project Screen**

Upon launching the IDE you will be presented with a welcome dialog.



This will give you quick access to the following options:

* **Record a new test in a new project**
* **Open an existing project**
* **Create a new project**
* **Close Selenium IDE**



# **Managing Tests and Suites in Selenium IDE**

## Adding a New Test

Click the + symbol at the top of the left side-bar menu, to the right of the Tests heading.

Provide a name for the test.

Click ADD.

You can either input commands manually or click the record icon in the top-right of the IDE to start recording.

## Grouping Tests into Suites

**Tests can be grouped together into suites.**

Upon project creation, a Default Suite is created automatically and your first test is added to it.

## Managing Test Suites

To access the Test suites panel, click the drop-down at the top of the left side-bar menu (e.g., click on the word Tests) and select Test suites.

* **Adding a Suite**
  + Click the + symbol at the top of the left side-bar menu, to the right of the Test Suites heading.
  + Provide a name for the suite.
  + Click ADD.
* **Adding a Test to a Suite**
  + Hover over the suite name.
  + Click the icon that appears to the right of the Test Suites heading.
  + Click Add tests.
  + Select the tests you want to add from the menu.
  + Click Select.
* **Removing a Test from a Suite**
  + Hover over the test name.
  + Click the X that appears to the right of the name.
* **Remove a Suite:**
  + Click the icon that appears to the right of the suite name.
  + Click Delete.
  + Confirm by clicking Delete again when prompted.
* **Rename a Suite:**
  + Hover over the suite name.
  + Click the icon that appears to the right of the name.
  + Click Rename.
  + Update the name.
  + Click RENAME.

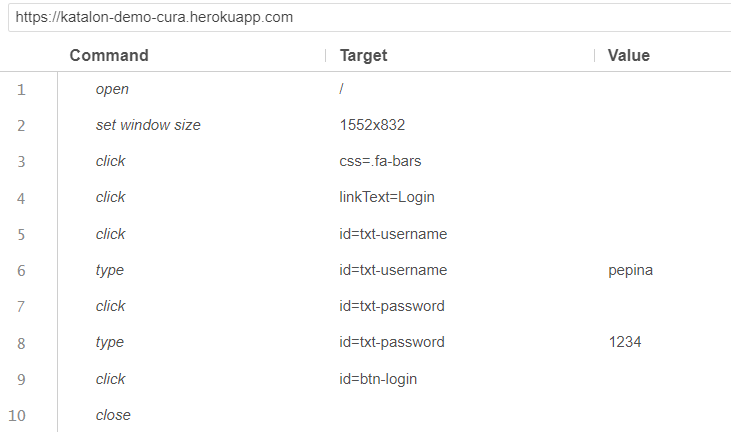
# **Creating Your First Script**

## Recording

**Record a script for an incorrect login attempt on a website**

* Open Selenium IDE.
* Choose "Record a new test in a new project" or "Create a new project". (The only difference is that if you choose the first option, you will be asked to name your project, provide an URL, and start recording directly and the second option creates an empty project.) In this demo we chose "Create a new project".
* Enter the URL of the page to be tested ([**https://katalon-demo-cura.herokuapp.com/**](https://katalon-demo-cura.herokuapp.com/)) in the base URL field.
* Press the record button. Selenium is now ready to record interactions with the browser.
* Click the menu icon on the website.
* Observe the message at the bottom indicating that Selenium registered the action.
* From the menu, select the login option.
* Fill in the username and password fields.
* Press the login button.
* Close the browser.
* Return to Selenium IDE.
* Stop the recording.

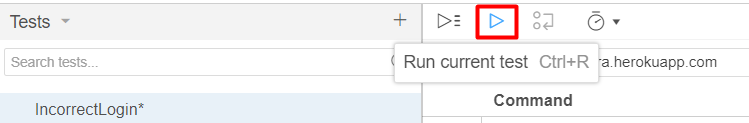
**The entire script for the incorrect login is now ready.**



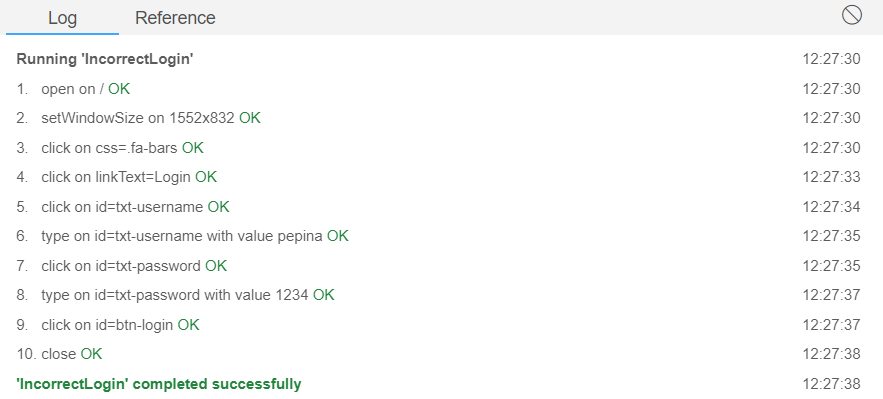
## Playback

**Run current test.**

* On the IDE's toolbar menu, click the "Run Current Test" button.



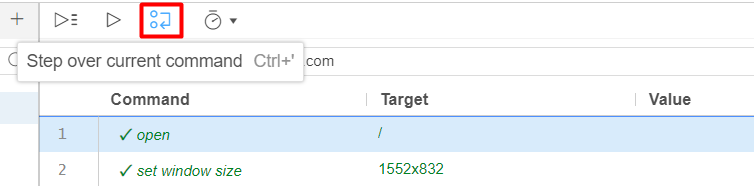
* Selenium IDE will execute all the recorded interactions.
* The Log Reference pane will display an overview of the script execution, showing each step's results.



## Debugging

**Run the test in debug mode.**

Note: In order to start the test in debug mode you should step on the first command.



**Review Commands:**

* **Open Command:** Opens the specified page.
* **Set Window Size Command:** Sets the size of the browser window.
* **Click Command:** Clicks on the menu icon.
* **Select Login Option:** Selects the login option from the menu. Redirects to the login page.
* **Press Username Field:** Clicks on the username field.
* **Type Command (Username):** Enters the value "pepina" into the username field.
* **Press Password Field:** Clicks on the password field.
* **Type Command (Password):** Enters the value "1234" into the password field.
* **Click Login Button:** Presses the login button.
* **Close Browser:** Closes the browser.

**Observe Test Results:**

* Verify that all commands have passed correctly (commands turn green).
* If a command fails, it will turn red, indicating where something went wrong.
* Check the logs for detailed information about each step taken and the test results.

## Refactoring

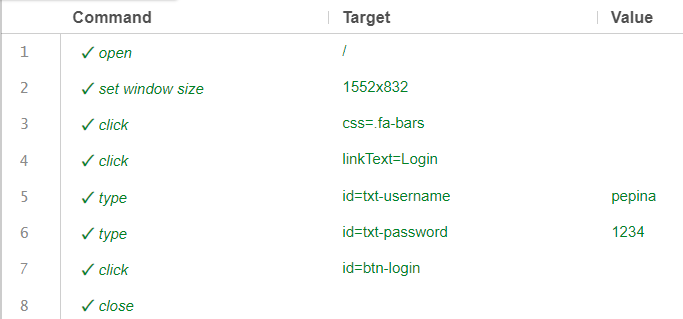
Sometimes the recorder captures additional, unnecessary steps that don't contribute to the main test flow. These extra steps, often referred to as "parasitic steps," can clutter your test script and may need to be removed to streamline the test case.

**Identify Unnecessary Steps:**

* **Click Username Field:** Remove this command. The type command will automatically click on the field and enter the value.
* **Click Password Field:** Remove this command. The type command will automatically click on the field and enter the value.

**Run the Refactored Script:**

* Execute the script to ensure it still works as intended.
* Verify that all steps pass and the test behaves as expected.



## Saving

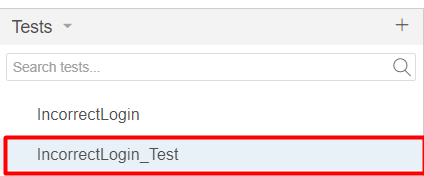
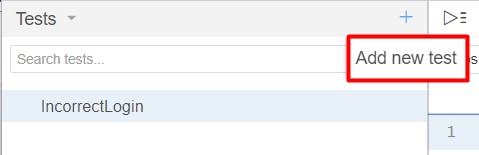
* Click the save button located in the upper right corner of the menu bar.
* It will prompt you to for a location and name of where to save the project.
* The end result is a single file with .side extension.

# **Creating Your First Test**

Recording alone is not enough. We have to implement assertion commands. It is up to us to select the appropriate elements for the assertion and determine the expected result or value. Assertions in our "InvorrectLogin" script can be put in two ways. The first way is to add an assertion while recording. The second way is to add an assertion manually.

## Add assertions while recording

**Let's create a new test in which we record the same steps and dynamically add assertions.**



**Add assertions to check the home page main text**, assert the **text on the login page**, and assert the **error message** for incorrect login.

* Click the record button in Selenium IDE.
* Right-click on the main text element "CURA Healthcare Service" where the assertion is to be placed.
* Hover over the additional Selenium-related option in the context menu.
* Hover over "assert" and select "text" to place a text assertion.

This will generate an assertText command with the locator of the element and the expected text value. This assertion ensures that Selenium is on the correct website.

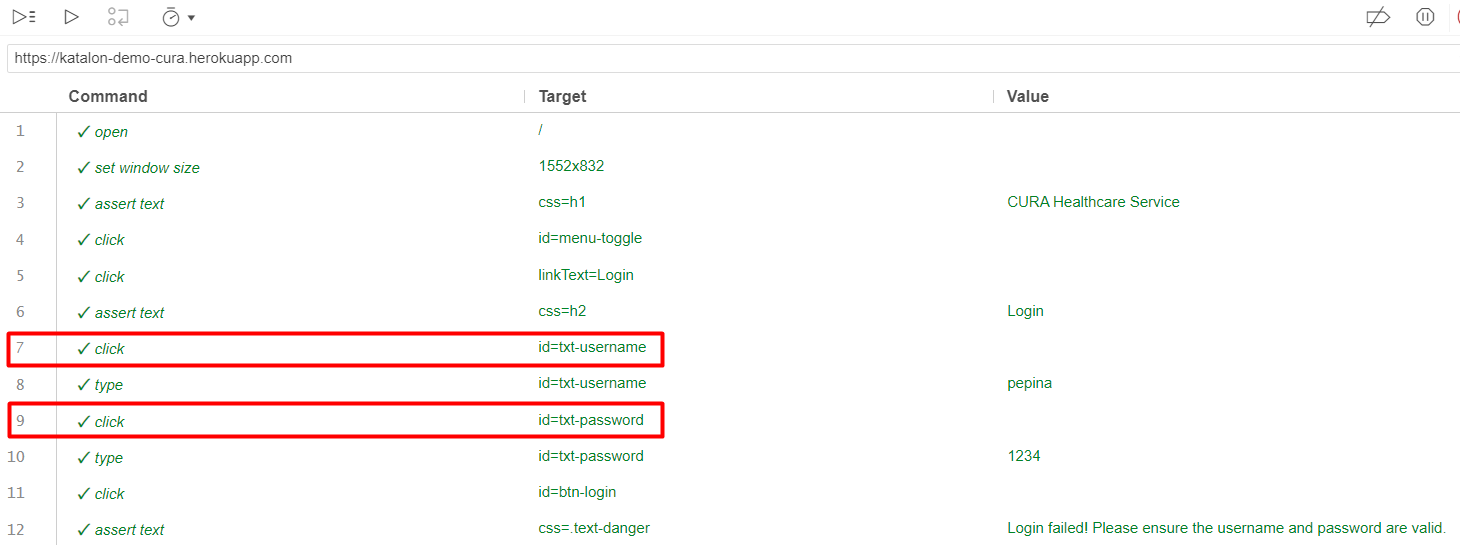
* Continue recording and perform actions to navigate to the login page.
* Right-click on the "Login" text element on the login page.
* Hover over "assert" and select "text".

This will add another assertText command to ensure the "Login" text on login page is displayed correctly.

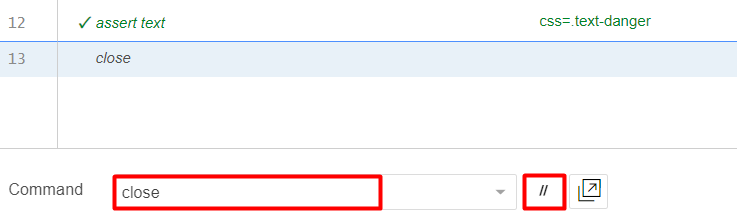
* Fill in the username and password fields.
* Press the login button.
* After the login attempt, add an assertion to check for the error message.
* Right-click on the error message element.
* Hover over "assert" and select "text".

This will add an assertText command to verify that the error message is displayed after entering incorrect login details.

* Turn off the recording in Selenium IDE.
* Refactor the script like we did in the previous exercise.



As you can see, we are missing the close command at the end. Let's add it manually.

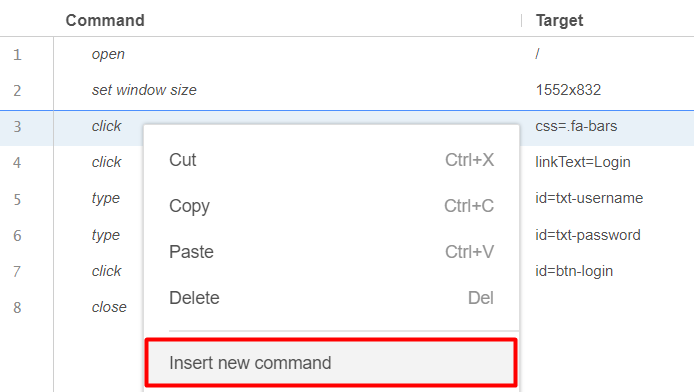


If you don't want Selenium IDE to open a new browser every time you run the test, you can disable the command by pressing the two slashes. When you want to activate the command again, just press the same button again.

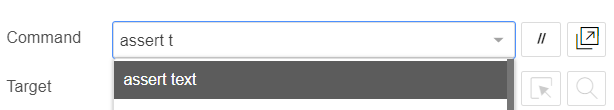
## Add assertions manually

**Head back to the first script that we wrote.**

* Right-click on the command in step 3 (after the browser opens and its size is set).
* Select "Insert new command" from the context menu. A new step will appear above the selected step.

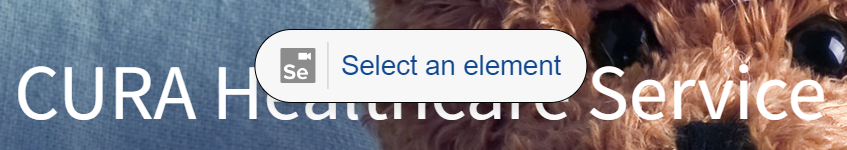


* In the new command, enter assert text.



* Locate the item on the page you want to assert. For example, if you want to verify the presence of a header, locate that header. It will show in the Target field.

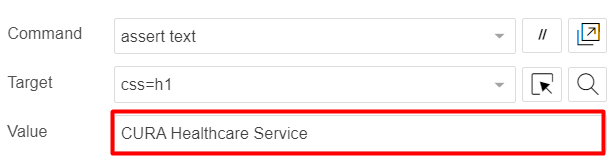






* Copy the text from the located element and paste it into the value field of the new command.





**Follow the steps to add second and third assertions by yourself**

* **Add the Second Assertion (Login Text)**
* Right-click on step 6 (just after navigating to the login page).
* Select "Insert new command".
* Enter assert text in the new command.
* Locate the login text element on the page.
* Copy the text from the located element and paste it into the value field of the new command.
* **Add the Third Assertion (Error Message)**
* Enter wrong login credentials to force an error message to appear.
* Right-click on step 10 (where the wrong credentials are submitted).
* Select "Insert new command".
* Enter assert text in the new command.
* Enable **developer tools by pressing F12**.
* In the **developer tools**, use the element selector to locate the error message element.
* Copy the text from the element in the developer tools and paste it into the value field of the new command.

**Note:** **Copying from developer tools can be more precise** as it includes any trailing spaces that might be missed when copying directly from the page.

# **Selenium IDE Advanced**

## Adding Conditional Logic

You can write your script from scratch, or you can duplicate one of the scripts that we already wrote, rename it, and delete all the commands except for the first two. Then:

* **Add store title Command**
* Set the Command to store title.
* Set the Value to webpageTitle.
* **Add if Command**
* Set the Command to if.
* Set the Target to ${webpageTitle} === "CURA Healthcare Service".
* **Add echo Command for Matched**
* Set the Command to echo.
* Set the Target to Matched.
* **Add else Command**
* Set the Command to else.
* **Add echo Command for Unmatched**
* Set the Command to echo.
* Set the Target to Unmatched.
* **Add end Command**
* Set the Command to end.

## \* Looping

Following the examples in the presentation, try these by yourself:

* **times**
* **do**
* **while**
* **forEach**