

Selenium Windows and Tabs Handling

Handling **Windows and Tabs in Selenium** is an important part of automation testing when dealing with multiple browser windows or tabs, such as popups, authentication windows, or links that open in new tabs.

◆ Key Concepts

- `driver.getWindowHandle()` → Gets the **current window handle** (a unique ID for the tab/window).
- `driver.getWindowHandles()` → Returns a **set of all window handles**.
- `driver.switchTo().window(handle)` → Switches control to a different tab/window using its handle.
- `driver.close()` → Closes the current tab/window.
- `driver.quit()` → Closes all tabs/windows and ends the session.

✓ Example 1: Handle Two Tabs

🔧 Scenario:

Clicking a link opens a new tab, and we want to switch between the original tab and the new one.

✓ Code (with Explanation in Comments):

```
python

from selenium import webdriver
from selenium.webdriver.common.by import By
import time

# Step 1: Launch the browser
driver = webdriver.Chrome()

# Step 2: Open the main URL
driver.get("https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_win_open")
driver.maximize_window()

# Step 3: Switch to iframe if the button is inside an iframe
driver.switch_to.frame("iframeResult")

# Step 4: Click the button to open a new window/tab
driver.find_element(By.TAG_NAME, "button").click()

# Step 5: Get all window handles
handles = driver.window_handles
print("All Window Handles:", handles)

# Step 6: Switch to the new tab (last handle)
driver.switch_to.window(handles[1])
print("New tab title:", driver.title)

# Step 7: Wait and then close new tab
time.sleep(3)
driver.close()

# Step 8: Switch back to original tab
```

```
driver.switch_to.window(handles[0])
print("Original tab title:", driver.title)

# Optional: Close the browser
driver.quit()
```

✓ Example 2: Handle Multiple Tabs Dynamically

If you don't know which handle belongs to which tab, you can loop through them.

```
python

original = driver.current_window_handle
all_windows = driver.window_handles

for handle in all_windows:
    driver.switch_to.window(handle)
    if driver.title == "Expected Title":
        break # Found the tab we want
```

Summary

Operation	Selenium Code
Get current window handle	<code>driver.current_window_handle</code>
Get all handles	<code>driver.window_handles</code>
Switch to a tab/window	<code>driver.switch_to.window(handle)</code>
Close current tab	<code>driver.close()</code>
Close all tabs	<code>driver.quit()</code>

Pro Tips

- Always **wait** a bit (e.g., with `time.sleep()` or `WebDriverWait`) after opening a new tab before switching.
- To handle **popup blockers**, configure browser options properly.
- Use `driver.title`, `driver.current_url`, or element presence to identify which tab is active.

Would you like a **real-time mini-project** example where you open multiple tabs and extract data from each?

getWindowHandle() in Selenium

✓ Definition:

`driver.getWindowHandle()` returns a **unique identifier (handle)** for the **currently active browser window or tab**.

Each browser window or tab opened by Selenium WebDriver has a unique ID (called a "window handle") used to switch between them.

Example:

```
python

from selenium import webdriver

driver = webdriver.Chrome()
driver.get("https://example.com")

# Get the current window handle
current_handle = driver.current_window_handle

print("Current Window Handle:", current_handle)

driver.quit()
```

Explanation:

- The value returned looks like a long string (e.g., "CDwindow-1234567890ABCDEF").
- You use this handle to keep track of or return to a particular window/tab after switching.

Common Use Case:

```
python

original_handle = driver.current_window_handle

# Assume clicking this link opens a new tab
driver.find_element(By.LINK_TEXT, "Open new tab").click()

# Wait and switch to the new tab
all_handles = driver.window_handles

for handle in all_handles:
    if handle != original_handle:
        driver.switch_to.window(handle)
        break

# Do something in the new tab...

# Switch back to original
driver.switch_to.window(original_handle)
```

Related Methods

Method	Purpose
<code>getWindowHandle()</code>	Gets current window/tab's handle
<code>getWindowHandles()</code>	Gets handles for all open windows
<code>switch_to.window(hdl)</code>	Switches control to specific window/tab

Would you like a hands-on practice problem using `getWindowHandle()` and `getWindowHandles()`?