getWindowHanle and getWondowHandles

The methods getWindowHandle and getWindowHandles are part of the WebDriver interface in Selenium, which is a popular tool used for automating web browsers. These methods are used to handle multiple browser windows or tabs.

**getWindowHandle():**

The getWindowHandle method is used to retrieve the window handle of the current browser window or tab. A window handle is a unique identifier assigned to each window or tab opened by the WebDriver.

It returns a string value representing the window handle of the current window.

This method is typically used when you want to switch between different windows or perform actions on a specific window.

**getWindowHandles():**

The getWindowHandles method returns a set of window handles for all the currently open browser windows or tabs.

It returns a set of strings, with each string representing a window handle.

This method is useful when you need to iterate through multiple windows or tabs, such as closing or switching between them.

public class WindowHandleExample {

public static void main(String[] args) {

// Set the path to the chromedriver executable

System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

// Create a new instance of the ChromeDriver

WebDriver driver = new ChromeDriver();

// Open a URL in the browser

driver.get("https://example.com");

// Get the window handle of the current window

String currentWindowHandle = driver.getWindowHandle();

System.out.println("Current window handle: " + currentWindowHandle);

// Open a new window or tab

driver.executeScript("window.open('https://google.com');");

// Get all the window handles

Set<String> windowHandles = driver.getWindowHandles();

System.out.println("Window handles: " + windowHandles);

// Switch to the newly opened window

for (String handle : windowHandles) {

if (!handle.equals(currentWindowHandle)) {

driver.switchTo().window(handle);

break;

}

}

// Perform actions on the new window

System.out.println("Current URL: " + driver.getCurrentUrl());

// Close the new window

driver.close();

// Switch back to the original window

driver.switchTo().window(currentWindowHandle);

// Perform actions on the original window

System.out.println("Current URL: " + driver.getCurrentUrl());

// Close the original window and quit the driver

driver.quit();

}

}