Rico-Kali Hayes
SDET AUG2023
08/03/2023

Selenium WebDriver Interview Questions and Answers

1. **What is Selenium Webdriver?**
   Selenium WebDriver is a web framework for automating web-based application testing. It allows us to verify that the front-end of a web application is performing as expected.

2. **What are different locators in Selenium WebDriver?**
   Selenium supports 8 different types of locators: **id**, **name**, **className**, **tagName**, **linkText**, **partialLinkText**, **CSS selector** and **xpath**.

3. **Which locators would you prefer to use for search for a web element?**
   I prefer using the **name**, **className**, **tagName** and **CSS Selector** locator methods when searching for web elements. However, using **id** is my most preferred because it seems to be more reliable when recognizing an element.

4. **What is the difference between implicit and explicit wait?**

- **Implicit wait** waits for an element to appear on the page. Implicit wait applies globally.
- **Explicit wait** waits for a specific condition, such as the presence of an element or the element to be clickable. Explicit wait applies locally to a specific element.

5. **What is Fluent wait in Selenium Webdriver?**
   Fluent Wait looks for a web element repeatedly at regular intervals until timeout happens or until the object is found. Fluent Wait commands are most useful when interacting with web elements that can take longer durations to load.

6. **What is difference between driver.quit() and driver.close() methods?**

- **close()** closes only the current window on which Selenium is running automated tests. The WebDriver session, however, remains active.
- **quit()** method closes all browser windows and ends the WebDriver session.

7. **What are different exceptions that can be used in explicit wait?**

- **A couple exceptions that the Explicit wait** can throw are
  - **TimeoutException** if the element doesn't meet an expected condition within the specified timeout.
  - **ElementNotVisibleException** if the specified timeout is exceeded for an element to become visible.

8. **How do you handle alerts and pop-ups in Selenium Webdriver?**

- We can handle alerts with the following methods:
  - **dismiss():** This method is used when the ‘Cancel’ button is clicked in the alert box.
  - **accept():** This method is used to click on the ‘OK’ button of the alert.
  - **getText():** This method is used to capture the alert message.
  - **sendKeys(String stringToSend):** This method is used to send data to the alert box.

- We can handle pop-ups with the following methods:
    - **getWindowHandles():** In order to handle the opened windows by Selenium Webdriver, you can use Driver.getWindowHandles() to switch between the windows.
    - **Driver.getWindowHandle()**:When the webpage is loaded, you can handle the main window by using driver.getWindowHandle(). It will handle the current window that uniquely identifies within the driver instance.

9. How do you scroll a page in Selenium WebDriver?
We can scroll different directions of a webpage by using the **window.scrollBy()** method of the **JavascriptExecutor** interface. We pass the pixel values (negative or positive) to be traversed along the x and y axis inside the scrollBy() method signature.

```
JavascriptExecutor j = (JavascriptExecutor)driver;
j.executeScript("window.scrollBy(0,500)");
j.executeScript("window.scrollBy(0,-500)");
```

10. How do you select a value from dropdown in Selenium Webdriver?
- To select an *option* with its value we have to use the **selectByValue()** method and pass the value attribute of the option that we want to select as a parameter to that method.

```
WebElement v = driver.findElement(By.name("selt"));
Select s = new Select(v);
s.selectByValue("val1");
```

11. How do you mouse hover over a web element and click on an element in Selenium Webdriver?
- Use the moveToElement method of the Actions class to move the mouse cursor over the element you want to hover. To perform the click action, use the click method of the Actions class.

  Ex. **WebElement** element = *driver.findElement*(**By**.*id*(“element-id”));
  Actions actions = new **Actions**(driver);
  actions.*moveToElement*(element).*perform*();
  actions.*click*(element).*perform*();

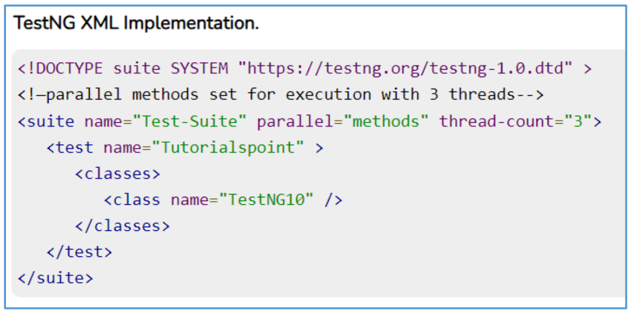
12. What is Test Flakiness in automation?
- A test that intermittently fails for no apparent reason, or works in your local machine but fails with continuous integration. Flaky tests hinder development, slow down progress, and hide design problems.

13. What is Page Object Model?
- Page Object Model is an object design pattern where webpages are represented as classes, and the various elements on the page are defined as variables in the class. All possible user interactions can then be implemented as methods of the classes:
    - *Advantages*: Easy maintenance, code reusability, readability, and reliability of test scripts.

14. Please give me an example of page object model? Have you ever created page object model framework from scratch?
- An example of a page object model is that a class would be created for each corresponding html document it refers to. Each of these classes will contain variables and methods that represent UI elements or Objects present in the web page. It will also contain operations to be performed on each element.

15. How do you preserve the session in Page Object Model?

- We can perform session handling with the TestNG framework. A TestNG execution configuration is done in the TestNG XML. We create multiple sessions by using the attributes – parallel and thread-count in the XML file. The attribute thread-count controls the number of sessions to be created while executing the tests in a parallel mode.



TestNG XML Implementation.

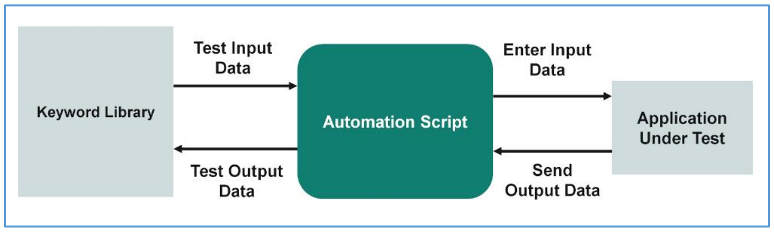
```
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd" >
<!–parallel methods set for execution with 3 threads-->
<suite name="Test-Suite" parallel="methods" thread-count="3">
   <test name="Tutorialspoint" >
      <classes>
         <class name="TestNG10" />
      </classes>
   </test>
</suite>
```

16. What is Data Driven framework?

- A Data Driven Framework in Selenium is a technique of separating the "data set" from the actual "test case." Data Driven framework is used to drive test cases and suites from an external data source. The data source can be data sheets like xls, xlsx, and csv files.

17. What is Keyword Driven framework?

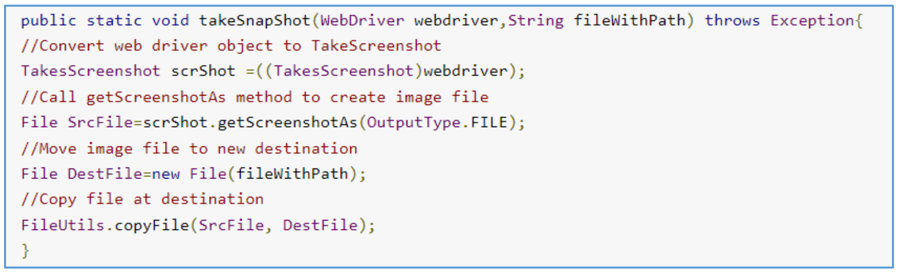
- A keyword driven framework in Selenium is a testing approach that uses a table of keywords to represent the actions and inputs for each test case. All actions and instructions are written in a external file. Test cases are then constructed by calling the keywords in a specific sequence to perform the desired testing steps.



a. The automation script will read the instructions or test input data from the Excel sheet.
b. The input data is entered in the application under test.
c. The test cases are performed and results are returned.
d. The test output data is stored in the Excel sheet.

18. How do you take screenshot in Selenium Webdriver?

- To capture screenshots in Selenium,
  - Convert web driver object to TakeScreenshot.
  - Call getScreenshotAs method to create image file.
  - Copy file to Desired Location.



```
public static void takeSnapShot(WebDriver webdriver,String fileWithPath) throws Exception{
//Convert web driver object to TakeScreenshot
TakesScreenshot scrShot =((TakesScreenshot)webdriver);
//Call getScreenshotAs method to create image file
File SrcFile=scrShot.getScreenshotAs(OutputType.FILE);
//Move image file to new destination
File DestFile=new File(fileWithPath);
//Copy file at destination
FileUtils.copyFile(SrcFile, DestFile);
}
```

19. How do you upload a file using Selenium Webdriver?

- We can upload files in Selenium by using the **sendKeys()** method.

```
WebElement upload_file = driver.findElement(By.xpath("//input[@id='file_up']"));

upload_file.sendKeys("C:/Users/Sonali/Desktop/upload.png");
```

20. What is XPATH? How do you use Xpath in Selenium Webdriver?

- **XPath** (XML Path Language) is an expression language. It uses path expressions to select nodes or node-sets in a web document.

21. What is absolute and relative xpath?

- ***Absolute Xpath*:** It contains the complete path from the Root Element to the desire element.
- ***Relative Xpath*:** The relative Xpath starts from the middle of the DOM Structure and is not as long as the absolute XPath. The Relative Path is used for testing of an element.

22. What is the difference between single and double slash in Xpath?

- The single forward slash **(/)** selects only the immediate child elements, while the double forward slash **(//)** selects all descendants of the current node.

23. How do you handle Ajax calls in Selenium Webdriver?

- One way of handling an Ajax call is by using the explicit **wait.Until()** method from the ***JavascriptExecutor*** interface. It can wait for the Ajax call to complete, then return true once it has finished. Which can allow us to take further action based on the condition.

24. What are assertions in Selenium Webdriver? What are different types of assertions?

- Asserts are validations or checkpoints for an application. Assertions validate that an application's behavior is working as expected and used in automated test cases to help testers understand if tests have passed or failed.
- Selenium Assertions can be of three types:
  - **assert:** When an "assert" fails, the test is aborted.
  - **verify:** When a "verify" fails, the test will continue execution, logging the failure.
  - **waitFor**: A "waitFor" command waits for some condition to become true.

25. What test cases are best candidates for automation? Why?

- Tests that are best for automation are ones that run frequently and require large amounts of data to perform repetitive actions. These tests include:
  - **Unit Tests**
  - **Functional Tests**
  - **Regression Tests**
  - **Smoke Tests**
  - **Performance Tests**