SELENIUM FAQ'S

Part-3

Ques.51. How to verify tooltip text using selenium?

- Ans. Webelements have an attribute of type 'title'. By fetching the value of 'title' attribute we can verify the tooltip text in selenium.

```
String toolTipText = element.getAttribute("title");
```

Ques.52. How to locate a link using its text in selenium?

- Ans. Using **linkText()** and **partialLinkText()** we can locate a link.
- The difference between the two is linkText matches the complete string passed as parameter to the link texts. Whereas partialLinkText matches the string parameter partially with the link texts.

```
WebElement link1 = driver.findElement(By.linkText(“pavantestingtools"));
WebElement link2 = driver.findElement(By.partialLinkText(“testingtools"));
```

Ques.53. What are DesiredCapabilities in selenium webdriver?

- Ans. Desired capabilities are a set of key-value pairs that are used for storing or configuring browser specific properties like its version, platform etc in the browser instances.

Ques.54. How can we find all the links on a web page?

- Ans. All the links are of anchor tag 'a'. So by locating elements of tagName 'a' we can find all the links on a webpage.

```
List<WebElement> links = driver.findElements(By.tagName("a"));
```

Ques.55. What are some commonly encountered exceptions in selenium?

- Ans. Some of the commonly seen exception in selenium are-
- NoSuchElementException - When no element could be located from the locator provided.
- ElementNotVisibleException - When element is present in the dom but is not visible.
- NoAlertPresentException - When we try to switch to an alert but the targetted alert is not present.
- NoSuchFrameException - When we try to switch to a frame but the targetted frame is not present.
- NoSuchWindowException - When we try to switch to a window but the targetted window is not present.
- TimeoutException - When a command execution gets timeout.
- InvalidElementStateException - When the state of an element is not appropriate for the desired action.
- NoSuchAttributeException - When we are trying to fetch an attribute's value but the attribute is not correct
- WebDriverException - When there is some issue with driver instance preventing it from getting launched.

Ques.56. How can we capture screenshots in selenium?

- Ans. Using getScreenshotAs method of TakesScreenshot interface we can take the screenshots in selenium.

```
File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
FileUtils.copyFile(scrFile, new File("D:\\testScreenShot.jpg"));
```

Ques.57. How to handle dropdowns in selenium?

- Ans. Using Select class-

```
Select countriesDropDown = new Select(driver.findElement(By.id("countries")));
dropdown.selectByVisibleText("India"); //or using index of the option starting from 0
dropdown.selectByIndex(1); //or using its value attribute

dropdown.selectByValue("Ind");
```

Ques.58. How to check which option in the dropdown is selected?

- Ans. Using isSelected() method we can check the state of a dropdown's option.

```
Select countriesDropDown = new Select(driver.findElement(By.id("countries")));
dropdown.selectByVisibleText("India"); //returns true or false value

System.out.println(driver.findElement(By.id("India")).isSelected());
```

Ques.59. How can we check if an element is getting displayed on a web page?

- Ans. Using isDisplayed method we can check if an element is getting displayed on a web page.

```
driver.findElement(By locator).isDisplayed();
```

Ques.60. How can we check if an element is enabled for interaction on a web page?

- Ans. Using isEnabled method we can check if an element is enabled or not.

```
driver.findElement(By locator).isEnabled();
```

Ques.61. What is the difference between driver.findElement() and driver.findElements() commands?

- Ans.
- findElement() returns a single WebElement (found first) based on the locator passed as parameter. Whereas findElements() returns a list of WebElements, all satisfying the locator value passed.
- Syntax of findElement():
  WebElement textbox = driver.findElement(By.id("textBoxLocator"));
- Syntax of findElements():
  List <WebElement> elements = element.findElements(By.id("value"));
- Another difference between the two is- if no element is found then findElement() throws NoSuchElementException whereas findElements() returns a list of 0 elements.

Ques.62. Explain the difference between implicit wait and explicit wait.?

- Ans. An implicit wait, while finding an element waits for a specified time before throwing NoSuchElementException in case element is not found. The timeout value remains valid throughout the webDriver's instance and for all the elements.

```
driver.manage().timeouts().implicitlyWait(180, TimeUnit.SECONDS);
```

- Whereas, Explicit wait is applied to a specified element only-

```
WebDriverWait wait = new WebDriverWait(driver, 5);
wait.until(ExpectedConditions.presenceOfElementLocated(ElementLocator));
```

Ques.63. How can we handle window UI elements and window POP ups using selenium?

- Ans. Selenium is used for automating Web based application only(or browsers only). For handling window GUI elements we can use AutoIT or Sikuli.

Ques.64. What is Robot API?

- Ans. Robot API is used for handling Keyboard or mouse events. It is generally used to upload files to the server in selenium automation.

```
Robot robot = new Robot(); //Simulate enter key action
robot.keyPress(KeyEvent.VK_ENTER);
```

Ques.65. How to do file upload in selenium?

- Ans. File upload action can be performed in multiple ways-
  - Using element.sendKeys("path of file") on the webElement of input tag and type file i.e. the elements should be like…

```
<input type="file" name="fileUpload">
```

  - Using Robot API.
  - Using AutoIT.
  - Using Sikuli

Ques.66. How to handle HTTPS website in selenium? or How to accept the SSL untrusted connection?

- Ans. Using profiles in firefox we can handle accept the SSL untrusted connection certificate. Profiles are basically set of user preferences stored in a file.

Firefox

```
FirefoxProfile profile = new FirefoxProfile();
profile.setAcceptUntrustedCertificates(true);
profile.setAssumeUntrustedCertificateIssuer(false);
WebDriver driver = new FirefoxDriver(profile);
```

IE

```
DesiredCapabilities capabilities = new DesiredCapabilities();
capabilities.setCapability(CapabilityType.ACCEPT_SSL_CERTS, true);
System.setProperty("webdriver.ie.driver","IEDriverServer.exe");
WebDriver driver = new InternetExplorerDriver(capabilities);
```

Chrome

```
DesiredCapabilities handlSSLErr = DesiredCapabilities.chrome ()
handlSSLErr.setCapability (CapabilityType.ACCEPT_SSL_CERTS, true)
WebDriver driver = new ChromeDriver (handlSSLErr);
```

Ques.67 How to do drag and drop in selenium?

- Using Action class, drag and drop can be performed in selenium. Sample code-

```
Actions act = new Actions(driver);

act.clickAndHold(source Element).moveToElement(target Element).release().build().perform();

OR

act.dragAndDrop(source Element, target Element).build().perform();
```

Ques.68. How to execute javascript in selenium?

- Ans. JavaScript can be executed in selenium using JavaScriptExecuter. Sample code for javascript execution-

```
JavascriptExecutor js = ((JavascriptExecutor) driver);
js.executeScript("{Java script code }");
```

Ques.69. How to handle alerts in selenium?

- Ans. In order to accept or dismiss an alert box the alert class is used. This requires first switching to the alert box and than using accept() or dismiss() command as the case may be.

```
Alert alert = driver.switchTo().alert(); //To accept the alert
alert.accept();

Alert alert = driver.switchTo().alert(); //To cancel the alert box
alert.dismiss();
```

Ques.70. What is HtmlUnitDriver?

- Ans. HtmlUnitDriver is the fastest WebDriver. Unlike other drivers (FireFoxDriver, ChromeDriver etc), the HtmlUnitDriver is non-GUI, while running no browser gets launched.

Ques.71. How to handle hidden elements in Selenium webDriver?

- Ans. Using javaScript executor we can handle hidden elements-

- (JavascriptExecutor(driver)) .executeScript("document.getElementsByClassName(ElementLocator).click();");

Ques.72. What is Page Object Model or POM?

- Ans. Page Object Model(POM) is a design pattern in selenium.
  POM helps to create a framework for maintaining selenium scripts.
- In POM for each page of the application a class is created having the web elements belonging to the page and methods handling the events in that page.
- The test scripts are maintained in separate files and the methods of the page object files are called from the test scripts file.

Ques.73. What are the advantages of POM?

- Ans. The advantages are POM are-
- Using POM we can create an Object Repository, a set of web elements in separate files along with their associated functions. Thereby keeping code clean.
- For any change in UI(or web elements) only page object files are required to be updated leaving test files unchanged.
- It makes code reusable and maintainable.

Ques.74. What is Page Factory?

- Ans. Page factory is an implementation of Page Object Model in selenium. It provides @FindBy annotation to find web elements and PageFactory.initElements() method to initialize all web elements defined with @FindBy annotation.

```
public class SamplePage
{
WebDriver driver;
@FindBy(id="search")
WebElement searchTextBox;
@FindBy(name="searchBtn")
 WebElement searchButton;

//Constructor public samplePage(WebDriver driver)
{
this.driver = driver; //initElements method to initialize all elements
PageFactory.initElements(driver, this);
}
 //Sample method
public void search(String searchTerm)
{
 searchTextBox.sendKeys(searchTerm); searchButton.click();
}
}
```

Ques.75. What is an Object repository?

- Ans. An object repository is centralized location of all the objects or WebElements of the test scripts.
- In selenium we can create object repository using Page Object Model and Page Factory design patterns.