**SELENIUM**

**key Notes by Deepak Dara**

1. java collections (list and set) (iterator, hasNext(), next(), size())
2. looping statements- for loop, foreach loop and while
3. Exception handling- try catch, throws
4. Methods of String class.
5. OOPS concepts-encapsulation, inheritance, abstraction, polymorphism.

**Code optimization:-**

The process of reducing number of statements but still getting the same output is called as code optimization. Example:-

int i = 10; int j = 20; int k = i+ j;

System.out.println(k);

*OR*

int i = 10; int j = 20;

System.out.println(i+j);

*OR*

int i = 10; System.out.println(i+20);

*OR*

System.out.println(10+20); Example:-

class Dog

{

String name = "Pinky";

}

Dog d1 = new Dog(); String n = d1.name; int l = n.length(); System.out.println(l);

OR

Dog d1 = new Dog(); String n = d1.name;

System.out.println(n.length()); OR

Dog d1 = new Dog(); System.out.println(d1.name.length());

OR

System.out.println((new Dog()).name.length());

class Dog

{

static String name = "Blacky";

}

String n = Dog.name; int l = n.length(); System.out.println(l);

OR

String n = Dog.name; System.out.println(n.length());

OR

System.out.println(Dog.name.length());

**Upcasting**:-

Class A

{

void testA()

{

System.out.println("A1");

}

void testB()

{

System.out.println("B1");

}

}

Class B extends A

{

void testC()

{

System.out.println("C2");

}

void testB()

{

System.out.println("B2");

}

}

B b1 = new B(); b1.testA();//o/p A1

b1.textB();//o/p B2 as B() is overridden in class B b1.testc();//o/p C2

A a1 = b1;

a1.testA();//o/p A1 a1.testB();//o/p B2

a1.testc();//compile time error as we cannot access the child class members using a parent class reference.

Converting child data type to parent data type is called as upcasting. Java supports auto upcasting (i.e A a1 = b1; here it is not needed to write A a1 = (A)b1)

After upcasting if we call any method that is present in both parent and child, then it will always execute method of child class (as it will be over ridden in child class).

After upcasting, we cannot call any method which is present only in the child class(it will be hidden).

interface A

{

public void testA();

}

class B implements A

{

public void testA()

{

System.out.println("A2");

}

}

A a1 = new B(); a1.testA();//o/p is A2

**Runtime polymorphism:-**

The method/object behaves differently during runtime in different situations. this is called as runtime polymorphism. to acheive runtime polymorphism, we should perform following steps:-

1. Inheritance.
2. Method over riding.
3. Upcasting.

Example:- interface A

{

public void testA();

}

class B implements A

{

public void testA()

{

System.out.println("B1");

}

}

class C implements A

{

public void testA()

{

System.out.println("C1");

}

}

class Demo

{

public static void main(String ar[])

{

}

void testDemo(A a1)

{

a1.testA();

}

}

In the above example, A is an interface which has an abstract method testA(), which is implemented in both class B and class C. The method testDemo(), accepts an argument of type A. Since A is parent for both B and C, for testDemo() method, we can pass the argument as an object of B class or an object of C class. Hence ouput of the testDemo() method, changes depending on the input argument which can be decided during runtime only. Example:- If we pass object of B class, we will get o/p as B1 else if we pass object of C class, the o/p will be C1.

**Selenium:-**

Selenium is a free and open source web application automation tool.

1. To use selenium for commercial purposes, we need not purchase any license. it can be free downloaded from "<http://docs.seleniumhq.org/download>"(selenium standalone server selenium-server-standalone-4.x.jar)
2. Selenium is open source i.e we can view/download/customize it. It is available at https://github.com/SeleniumHQ/selenium
3. Using selenium, we can automate web applications such as gmail, FB, linkedin,
4. Using selenium we can test the functionality of the application automatically.

Q. Can we do performance testing using selenium?

* + No. We need to integrate selenium with jmeter

Q. What are the flavors of selenium?

* + selenium core
  + Selenium ide
  + selenium rc- also called as selenium1
  + selenium webdriver-also called as selenium 2

Q. What is latest version of selenium?

* + 2.49.0

Q. Languages supported by selenium:-

* + Java, C#, Ruby, python, JavaScript, perl, PHP, Objective-C, haskell, R, dart, Tcl

Q. Platforms supported by selenium:-

All platforms (UNIX does not have a browser and thus selenium cannot be used)

*Assignments*:-

Browsers supported by selenium:-

Q. What type of test case we select for automation?

1. Performance test cases
2. Unit test cases
3. Smoke test cases
4. **Regression test cases**

*Answer*: - Regression test case.

Q. Do we automate negative scenarios?

* Yes.

Q. Which type of test cases we should automate first?

* (Smoke)Sanity test cases first have to be automated.

Q. Can we do 100% automation?

* No because, we may have manual interventions(example:- authentication through bio metric scanner, capturing product details from the barcode scanner, captcha, OTP, Swiping a debit/credit card by swiping, verification of audio/video clips)

https:/[/w](http://www.actitime.com/download.php)w[w.actitime.com/download.php](http://www.actitime.com/download.php) XAMP

Apache mysql php

**Installing selenium:-**

* Go to the required folder and create a folder called "BSSW5".
* Open eclipse-->file-->switch workspace, select other.
* Browse and select newly created folder and click OK.
* It will restart the eclipse.
* Create a java project with the name "Automation".
* Copy selenium server standalone jar file.
* Right click on java project present in the eclipse and select paste.
* Right click on selenium jar file which is present inside java project. Go to build path and select "Add to build path".
* Create a package with the name "<desired name>" under src.
* Then create a class with the name "Demo" and write the code as shown below and execute it.

package qualitythought;

import org.openqa.selenium.firefox.FirefoxDriver; public class Demo

{

public static void main(String[] args)

{

FirefoxDriver f = new FirefoxDriver(); f.close();

}

}

Q. How selenium performs the action on the browser?

* By calling the native methods of the browser.

Q. Which protocol is used by selenium to communicate with the browser?

* JSON Wire(Java Script Object Notation)

*Handling chrome browser*:-

1. Go to ["htt](http://chromedriver.storage.googleapis.com/index.html)p[://chromedriver.storage.googleapis.com/index.html?](http://chromedriver.storage.googleapis.com/index.html)".
2. Click on the latest folder.(as of now 2.20)
3. Download the required zip file.
4. Unzip the file which will create a file with the name "chromedriver.exe"
5. Copy paste into required location.
6. Write the code as shown below and execute it.

package qualitythought;

import org.openqa.selenium.chrome.ChromeDriver; public class Demo

{

public static void main(String[] args)

{

System.setProperty("webdriver.chrome.driver","D:\\Selenium\\Drivers\\chromedriver\_win32\\chrome driver.exe");

ChromeDriver c = new ChromeDriver(); c.close();

}

}

*Handling IE Browser*:-

1. go to latest folder and download "IEDriverServer\_win32\_2.48.0.zip"
2. Unzip the file which will create a file with name "IEDriverServer\_Win32\_2.48.0.exe"
3. Copy paste in required folder.
4. Write code as shown below and execute:-

package qualitythought;

import org.openqa.selenium.ie.InternetExplorerDriver; public class Demo

{

public static void main(String[] args)

{

System.setProperty("webdriver.ie.driver", "D:/Selenium/Drivers/IEDriverServer\_Win32\_2.48.0/IEDriverServer.exe");

InternetExplorerDriver i = new InternetExplorerDriver(); i.close();

}

**Note**:

}

*Before executing the code, manually open IE browser. Go to tools-->internet options.*

*Select the security tab.*

*Select "Enable protected mode" for all of internet, local intranet, trusted sites, and restricted sites. Also ensure that browser zoom level is 100%.(ctrl+0 shortcut)*

Q. Write a script to perform following steps on the user specified browser

1. Enter URL as [http://www.google.com](http://www.google.com/)
2. Get the title and print it.
3. Close the browser.

package qualitythought import java.util.Scanner;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver; import org.openqa.selenium.firefox.FirefoxDriver;

public class Demo

{

public static void main(String[] args)

{

Scanner s = new Scanner(System.in); System.out.println("Browser?"); String browser = s.next();

if(browser.equals("GC"))

{

System.setProperty("webdriver.chrome.driver","D:\\Selenium\\Drivers\\chromedriver\_win32\\chrome driver.exe");

}

else

{

}

}

ChromeDriver c = new ChromeDriver(); c.testBrowser();

FirefoxDriver f = new FirefoxDriver(); f.testBrowser();

public static void testBrowser(WebDriver driver)

{

driver.ge[t("ht](http://www.google.com/)tp[://w](http://www.google.com/)w[w.google.com](http://www.google.com/)"); String t = driver.getTitle(); System.out.println(t);

}

}

We do upcasting to achieve run time polymorphism, so that we can execute the same code(testBrowser()) on any browser.

**Architecture of selenium:-**

|  |  |  |
| --- | --- | --- |
| Bindings  java | WebDriver  API | Driver |
| Chrome driver |
| C# |  | IEDriver |
| Ruby |  |  |
| python etc |  |  |

* Drivers is with respect to browsers
* Bindings are with respect to language in which code is written.
* Selenium supports multiple languages. For each language we have respective client binding which communicate with WebDriver API.
* WebDriver API uses browser specific drivers (chrome driver, IE driver etc) to perform the actions.

**Architecture of webdriver:-**

SearchContext (interface)

^

| WebDriver(interface)

^

| RemoteWebDriver(class)

FirefoxDriver, ChromeDriver classes extend the RemoteWebDriver class.

SearchContext is the super most interfaces in selenium which is extended by WebDriver interface. All the methods of these 2 interfaces, are implemented in RemoteWebDriver class and over ridden in respective browser classes (FirefoxDriver, ChromeDriver etc).

1. To run the script on any browser, we use upcasting.
2. We can upcast the object of the browser to any of its parent such as RemoteWebDriver or WebDriver or SearchContext.
3. It is a best practice to upcast the object to the maximum level possible. (may be SearchContext). But we should ensure that none of the required features are hidden. So, with respect to selenium WebDriver architecture, the best level to upcast is WebDriver as shown below

WebDriver driver = new FirefoxDriver();

package qsp;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class Demo

{

public static void main(String[] args)

{

WebDriver driver = new FirefoxDriver(); driver.get(“[http://www.google.com”)](http://www.google.com/); Thread.sleep(2000);

driver.navigate().to[(“h](http://www.gmail.com/)tt[p://www.gmail.com](http://www.gmail.com/)”); Thread.sleep(2000);

driver.navigate().back(); Thread.sleep(2000); driver.navigate().forward(); Thread.sleep(2000); driver.navigate().refresh(); Thread.sleep(2000); driver.quit();

}

Using the navigate() method, we can move forward and back. Using get() we cannot perform back and forward actions.

Q. How to open the page without using get()?

* Use navigate().to(URL) to open the page.

Q. What is the difference between get() and navigate()?

* Using get(), we can only enter the URL whereas using navigateTo(), we can enter URL, select back, select forward and refresh the page.

Q. Write a script to open google.com and verify that title of the page is “Google”: WebDriver driver = new FirefoxDriver();

driver.get(“[http://www.google.com”)](http://www.google.com/); String title = driver.getTitle(); If(title.equals(“Google”))

{

}

else

{

}System.out.println(“Pass”);

System.out.println(“Fail”);

Write a script to open google.com and verify that it is redirected to google.co.in WebDriver driver = new FirefoxDriver();

driver.get(“[http://www.google.com”)](http://www.google.com/); String URL1 = driver.getTitle();

String URL2 = “google.co.in”; If(URL1.contains(URL2))

{

}

else

{

}

System.out.println(“Pass”);

System.out.println(“Fail”);

\*\*\*Practice the methods of String class.

**HTML:-**

1. Before performing any action such as clicking and typing, we should specify the element on which the action needs to be performed.
2. Anything present on the web page is called as element such as textbox, button, link, checkbox, listbox etc.
3. These elements are created using HTML.
4. HTML stands for hyper Text Markup Language
5. We can use notepad itself to write HTML code by saving it we should specify the extension as “.html”.

Example:- Open notepad and write code as shown below.

<html>

<body>

UN:<input type = “text” value = “admin”> PW:<input type = “password” value = “abc”>

<input type = “button value = “Login”><br>

<input type = “checkbox”>Remember

<a href = “[http://www.google.com”](http://www.google.com/)>Forgot pwd?</a>

</body>

</html>

Go to file and select save.

Navigate to required location and save the file with html extension. Double click on file which opens the page in browser.

Opening the html file in selenium

WebDriver driver = new FirefoxDriver();

Driver.get(“/Users/guruprasadsr/Desktop/Hi.html”);

*Components of WebElement:-*

Any element present in the web page contains following components.

1. Tag – Anything present after “<” Example: - html, body, input, a, br etc
2. Attribute: - Anything present after the tag till the “>” symbol. Example:- type, id, class, value, name etc
3. Text: - Anything present after the “>” symbol till the end of the tag. Example:- Forgot password?, Remember etc

*Example*: -

<a href=”https://www.google.com”/>Google</a>

a is the tag, href = https://www.google.com/ is the attribute and Google is the text.

**Locators:-**

To find the element present on the page, we use locators. In selenium there are 8 types of locators. All are static

methods of By class. Locators are specified as argument for findElement(). Return type of findElement() is WebElement. WebElement is an interface. Following are the selenium locators.

1. By.tagName(“str”)
2. By.id(“str”)
3. By.name(“str”)
4. By.className(“str”)
5. By.linkText(“str”)
6. By.partialLinkText(“str”)
7. By.cssSelector(“str”)
8. By.xpath(“str”)

If the specified locator is matching with multiple elements, then findElement() returns the address of the first matching element.

If the specified locator is not matching with any of the elements, then find Element() will throw NoSuchEelementException.

**CSS Classic Style Sheet:**

CSS Selector is an expression it has the following syntax tag[attributeName = ‘attributeValue’]

Example:-

<html>

<body>

</body>

</html>

UN:<input type = "text"> PWD:<input type = "password">

In the above web page to identify the password field, we cannot use id, name, className, linkText or partialLinkText because they are not available. We can use tagName but, it is duplicate with UN. In this scenario, we can use cssSelector

*Example*: -

WebDriver driver = new FirefoxDriver();

Driver.get(“files:///Users/guruprasadsr/Desktop/D-Drive/Selenium/java\_Tutorials/HTML/DemoUNPWD.html”); driver.findElement(By.tagname(“input”)).sendKeys(“abcd”);

driver.findElement(By.cssSelector(“input[type=’password’]”)).sendKeys(“xyz”);

If following error is displayed for sendKeys(), then change the compiler version of the eclipse to the latest version.

Error: -

The method sendkeys() in the type WebElement is not applicable for arguments (String). Solution: -

1. Right click on java project and select properties.
2. Click on java compiler and select latest version such as 1.7 or 1.8 under the ‘Compiler compliance level’.
3. Click on ‘Ok’

**Firebug and Firepath: -**

To inspect the element when the right clicking is disabled, we can use Firebug. It is an add on for Mozilla Firefox browser. To install it

1. Open the Firefox browser.
2. Go to tools and select add ons.
3. Search for firebug.
4. Click install button of firebug.
   * Go to the required webpage. Press f12 which opens firebug window. Click on inspect button. Then click on required element. In chrome and IE browsers, press f12 which opens developer tool bar. Click on

inspect button and then click on required element.

* + To write CSS expression, we can use a tool called firepath. To install it, go to tools add ons and search for firepath. Click on install button of firepath.
  + To write and check css expression, press f12 and click on firepath tab. Select CSS option.(by default it will be XPATH). Type the css expression and press enter, it will highlight the matching element.
  + To write CSS in Google chrome: -
  + Press f12. Then press control+f and type the css expression in the text box available at the bottom. Of the developer tool bar.

*Example*:-

<html>

<body>

</body>

</html>

FN:<input type = "text"> LN:<input type = "text">

In the above web page, to identify the last name field we cannot use id, name, className, linkText, partialLinkText locators because they are not present. We can use tagName or cssSelector but both are duplicate with first name field. In this situation, we can use xpath.

**xpath:-**

It is a path of an element in the html tree. Html tree:-

document

|html

|body

|input FN

|input LN

*Examples*: -

/html/body/input matches with FN and LN both

/html/body/input[1] matches with FN only

/html/body/input[2] matches with LN only

*Using xpath in selenium*: -

WebDriver driver = new FirefoxDriver();

driver.get(“files:///Users/guruprasadsr/Desktop/D-Drive/Selenium/java\_Tutorials/HTML/DemoFNLN.html”); driver.findElement(By.tagname(“input”)).sendKeys(“Guruprasad”);

driver.findElement(By.xpath(“/html/body/input[2]”)).sendKeys(“S”);

DemoTable.html

file:///Users/guruprasadsr/Desktop/D-Drive/Selenium/Java\_Tutorials/HTMLs/DemoTable.html

**Absolute xpath:**

Example: -

<html>

<body>

<table border = "1">

<tbody>

</body>

</html>

</tr>

<tr>

</tr>

</tbody>

<td>SQL</td>

<td>300</td>

<td>Java</td>

<td>400</td>

html tree html

|body

|table

|tbody

|tr

|tr

| |td SQL

| |td 300

|td Java

|td 200

1. If we specify the path of the element, from the beginning of the tree(html) till the element then it is called as absolute xpath.
2. While writing this xpath, we use single / which represents immediate child element.
3. We can use index in xpath which always starts from 1.
4. Index will be 2 if the tag of the element is same which is present under the same parent.

|  |  |
| --- | --- |
| Absolute xpath | Matching Element |
| /html/body/table/tbody/tr/td | SQL, 300, Java, 400 |
| /html/body/table/tbody/tr[1] | SQL,300 |
| /html/body/table/tbody/tr[1]/td[1] | SQL |
| /html/body/table/tbody/tr[1]/td[2] | 300 |
| /html/body/table/tbody/tr[2] | Java, 400 |
| /html/body/table/tbody/tr[2]/td[1] | Java |
| /html/body/table/tbody/tr[2]/td[2] | 400 |
| /html/body/table/tbody/tr/td[1] | SQL,java |
| /html/body/table/tbody/tr/td[2] | 300,400 |
| /html/body/table/tbody/tr[1]/td[1]|  /html/body/table/tbody/tr[2]/td[2] | SQL,400 |
| /html/body/table/tbody/tr[1]/td[1] |  /html/body/table/tbody/tr[1]/td[2] |  /html/body/table/tbody/tr[2]/td[2] OR  /html/body/table/tbody/tr[1]/td |  /html/body/table/tbody/tr[2]/td[2] | SQL,300,400 |

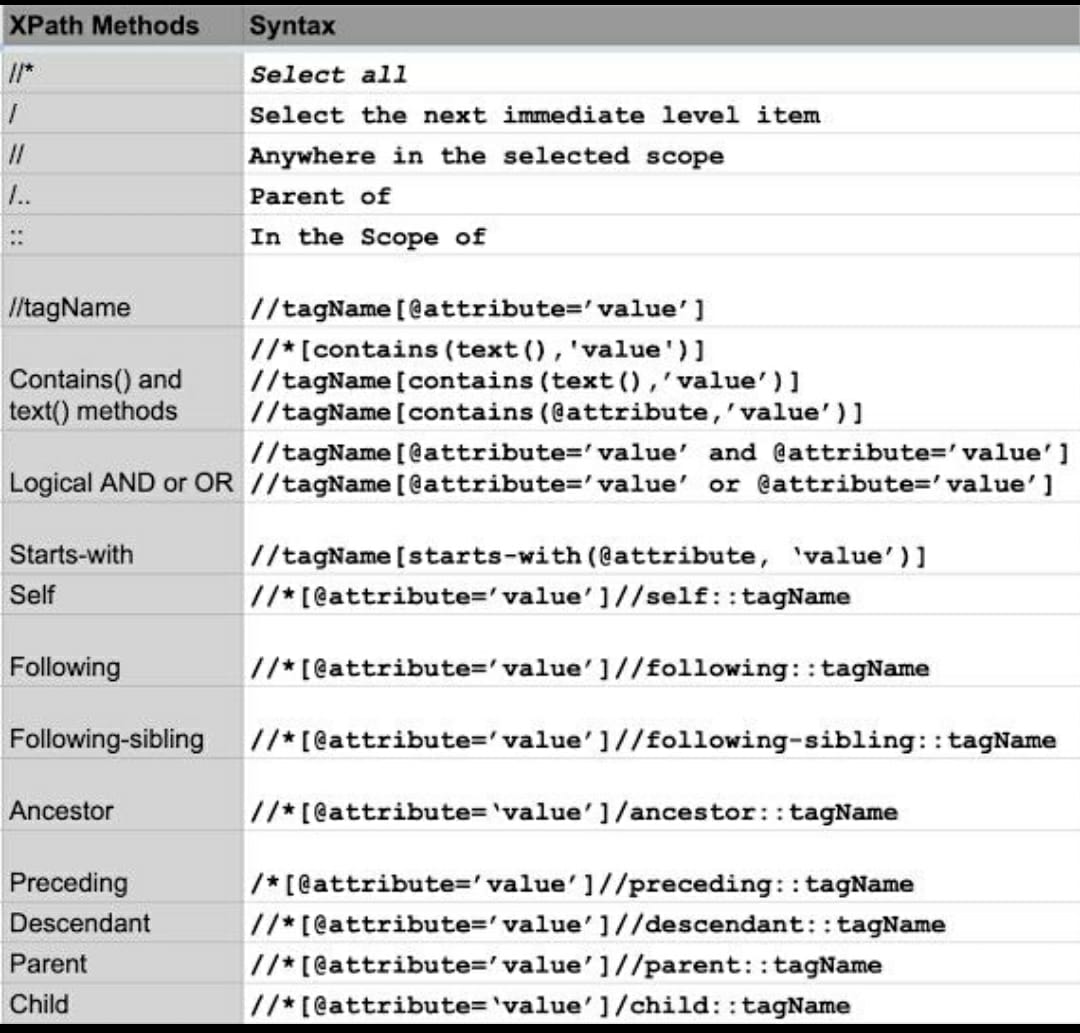
Writing absolute xpath on the elements which are present in the webpage will be very lengthy. To reduce the length, we use relative xpath.

**Relative xpath: -**

In relative xpath, we use ‘//’ which represents any child element (descendants)

|  |  |
| --- | --- |
| Relative xpath | Matching Element |
| //td | SQL, 300, Java, 400 |
| //tr[1]/td | SQL,300 |
| //tr[1]/td[1] | SQL |
| //tr[1]/td[2] | 300 |
| //tr[2]/td | Java, 400 |
| //tr[2]/td[1] | Java |
| //tr[2]/td[2] | 400 |
| //td[1] OR //tr/td[1] | SQL,java |
| //td[2] OR //tr/td[2] | 300,400 |
| //tr[1]/td[1] | //tr[2]/td[2] | SQL,400 |
| //tr[1]/td[1] |  //tr[1]/td[2] |  //tr[2]/td[2] OR  //tr[1]/td | //tr[2]/td[2] | SQL,300,400 |

Example:-

****

**Interview Questions: -**

Q. What is the difference between / and // ?

* / represent immediate child and // represents any child or descendant.

Q. Derive the xpath which matches with all the links present on the webpage?

* //a

Q. Difference between //a and //table//a ?

* //a matches with all the links which are present in the entire page. //table//a matches with all the links which are present inside all the tables.

Q. Write xpath which matches all the images and links present on the webpage?

* //img | //a

WebDriver driver = new FirefoxDriver();

Driver.get("file:///Users/guruprasadsr/Desktop/D-Drive/Selenium/Java\_Tutorials/HTMLs/DemoFNLN.html"); driver.findElement(By.xpath(“//input”)).sendkeys(“a”);

driver.findElement(By.xpath(“//input[1]”)).sendkeys(“b”); driver.findElement(By.xpath(“//input[2]”)).sendkeys(“c”);

When we get the driver object, the below are the methods that we can perform operation on a driver. In IDE like eclipse, when we enter driver. and click on space bar, it will show all the below methods.All these come under WebElement

1. get()
2. getCurrentUrl();
3. getTitle()
4. findElements()
5. findElement()
6. getPageSource()
7. close()
8. quit()
9. getWindowHandles()
10. getWindowHandle()
11. navigate()
12. manage()
13. switchTo()

**Other methods:**

1. getAttribute()
2. getLocation()
3. click()
4. clear()
5. Method Name :- get()

**Syntax:** get(url)

**Example:** driver.get();

**Purpose:** It will load a new web page in the current browser window. This is done using an http get operation, and the method will block until the load is complete.

**Parameters:** URL - The URL to load and it should be a fully qualified URL

1. Method Name: getCurrentUrl()

**Syntax:** getCurrentUrl()

**Example:** driver.getCurrentUrl();

**Returns:** The URL of the page currently loaded in the browser

1. Method Name: getTitle()

**Syntax:** getTitle()

**Example:** driver.getTitle();

**Purpose:** Gets the title of the current web page.

**Returns:** The title of the current page, with leading and trailing white space stripped, or null if one is not already set

1. Method Name: findElements()

**Syntax:** findElements(By by)

**Example:** driver.findElements(By.xpath("//");

**Purpose:** Find all elements within the current page using the given mechanism.

**Parameters:** By - The locating mechanism to use

**Returns:** A list of all WebElements, or an empty list if nothing matches

1. Method Name: findElement()

**Syntax:** WebElement findElement(By by)

**Example:** driver.findElements(By.xpath("//");

**Purpose:** Find the first WebElement using the given method.

**Parameters:** By - The locating mechanism

**Returns:** The first matching element on the current page Throws: NoSuchElementException - it will return exception if no matching elements are found

1. Method Name: getPageSource()

**Syntax:** getPageSource()

**Example:** driver.getPageSource();

**Purpose:** Get the source of the currently loaded page. If the page has been modified after loading (for example, by Javascript) there is no guarantee that the returned text is that of the modified page.

**Returns:** The source of the current page

1. **Method Name: close() Syntax:** void close() **Example:** driver.close();

**Purpose:** Close the current window, if there are multiple windows, it will close the current window which is active and quits the browser if it's the last window opened currently.

1. **Method Name: quit() Syntax:** void quit() **Example:** driver.quit();

**Purpose:** Quits this driver instance, closing every associated window which is opened.

1. Method Name: getWindowHandles()

**Syntax:** Set getWindowHandles()

**Example:** driver.getWindowHandles();

**Purpose:** Return a set of window handles which can be used to iterate over all the open windows of this Webdriver instance by passing them to switchTo().WebDriver.Options.window()

**Returns:** A set of window handles which can be used to iterate over all the open windows.

1. **Method Name: getWindowHandle() Syntax:** String getWindowHandle() **Example:** driver.getWindowHandle();

**Parameter:** Return an opaque handle to this window that uniquely identifies it within this driver instance. This can be used to switch to this window at a later date switchTo

WebDriver.TargetLocator switchTo() The next future commands will be performed to a different frame or window.

1. **Method Name: navigate() Syntax:** WebDriver.Navigation navigate() **Example:** driver.navigate.to("");

**Purpose:** An abstraction allowing the driver to access the browser's history and to navigate to a given URL.

**Returns:** A WebDriver.Navigation that allows the selection of what to do next

Click here to know more on Navigation methods

1. Method Name: manage()

**Syntax:** WebDriver.Options manage() **Purpose:** Gets the Option interface **Returns:** An option interface

1. switchTo()

**Syntax:** Webdriver.TargetLocator()switchto()

**Purpose:**Send future commands to a different frame or window.

**Returns:**A TargetLocator which can be used to select a frame or window

* 1. **Xpath by attribute:**

Using relative xpath we can reduce the length of expression but it may match with multiple elements even after using index.In order to identify the element uniquely we can use attribute in the xpath expression using following syntax:

**//tag[@AttributeName=’AttributeValue’] Example**: //input[@placeholder=’Username’]

**IQ 1]Can we use multiple attribute in the xpath expression Ans:**

Yes

**Example:1)** //input[@placeholder=’Username’ AND@name=’username’]

//input[@placeholder=’Username’ OR name=’username’]

**2)**//input[@value=’Log In’] **3)**//input[@id=’next’] **4)**//input[@value=’Next’]

* 1. **Xpath by text():**

If attribute is not present (or) attribute is matching with multiple elements in such cases we can use xpath by text,which has following syntax:

**//tag[text()=’textvalue’]**

**Example: 1)**//div[text()=’Login’] **2)**//div[text()=’USERS’]

**NOTE:** In the same xpath expression we can specify both attribute and text()

**HANDLING NON-BREAKABLE SPACE:**

1. developer can give the space in the value using spacebar or using Keyword **&nbsp**[non breakable space]
2. when we inspect the element in the browser we cannot make out whether &nbsp is used or not



1. If value has &nbsp then xpath will not identify such elements

**Example:**

<html>

<body>

<button type=”submit>&nbspOK&nbsp</button>

</body>

</html>

**IN FIREBUG:**

\*<button type=”submit”> OK </button> not identify the element

\*//button[text()=’ OK ‘] not identify the element To handle non breakable space we should use contains() which has followingsyntax:

//tag[contains(text(),’textvalue’)]

//tag[contains(@AttributeName,’AttributeValue’)]

**Example:**

* 1. attribute example button[contains(text(),’OK’)]

**Example:** //input[contains(@value,’Create Type of Work’)]login to actitime>settings>types of work>create types of work> create types of work>

* 1. text example

//a[contains(text(),’delete’)]

**NOTE:**

We use contains() if value has **non-breakable** space or if **value is keep changing**.

**Example:**//span[contains(text(),’Inbox’)]

**XPATH TRAVERSING**

We can derive a xpath expression which can navigate from one element to another element which is called as traversing.It supports 2 types of traversing;

1. Forward Traversing
2. Backward Traversing

**EXAMPLE:**

<html>

<body>

<table border=”1”>

<tbody>

<tr>

<td>1</td>

<td>Unix</td>

<td>300</td>

</tr>

<tr>

<td>2</td>

<td>Java</td>

<td>400</td>

</tr>

</tbody>

</table>

</body>

</html>

* 1. **FORWARD TRAVERSING:**

Navigating from parent element to any of its child element is called as forward traversing.

**Example:**

**Navigating from;**

table to Unix;

//table/tbody/tr[1]/td[2] Table to java;

//table/tbody/tr[2]/td[2]

* 1. **BACKWARD TRAVERSING:**

Navigating from child element to any of its parent element is called as backward traversing

**Example:**

Navigating from; Unix to table;

//td[text()=’Unix’]/../../..

Java to table;

//td[text()=’Java’]/../../..

**5) INDEPENDENT DEPENDENT XPATH:**

If the element is completely changing (or) it duplicate with some other elements we can use independent dependent concept of xpath to identify it.

**EXAMPLE:**

Derive xpath to identify cost of UNIX

**Step#1:**Inspect the independent element and Note down its source code.

**Step#2:**Place the mouse pointer on the source code of independent element and move the mouse pointer in upward direction step by step till it highlights both independent and dependent element. This will be the common parent ,add it to the html tree

**Step#3:**

Use arrow key to navigate till dependent element, add its path to the html tree.

**Step#4:**Using the tree derive the xpath expression which navigates from independent element to common parent and then to dependent element

<tr>

<td>Unix</td> td<td[3]

//td[text()=’Unix’]/../td[3]

**NOTE:**

While doing forward traversing we should navigate till end i.e, expand all the + sign.

**IQ2)Derive the xpath to match with download link of ruby present in download page of selenium**

Ans:

tr

<td>ruby</td> td[4]

a

//td[text()=’Ruby’]/../td[4]/a

**Note :**

The above xpath identifies the link which is present in 4th column only.To identify the download link even if column is keep changing we can use below xpath;

//td[text()=’Ruby’]/..//a[text()=’Download’]

**Example in actiTime:** //a[text()='Vidya']/../..//a[text()='set by default']

**Assignment:**

1. **Derive a xpath to identify the price of Mi 4i(Blue 16GB) present inflipkart**

//span[text()=' Rs. 11,998']

1. **Derive a xpath to identify add to compare checkbox of Redmi2 Prime(Grey 16 GB) in flipkart**

//input[@id='MOBE9T7GTHERTDAC']

1. **Derive a xpath to identify phone number of Mumbai present in isrtc.com**
2. **Identify help icon(?) present in actitimeapplication;**

(//div[@class=’popup menu arrow’])[3]

**Note:**

Sometimes we may not be able to identify the element even after using all types of xpath which is previously discussed.In such case we use GROUP INDEX(GI)

tbody

td(SQL) td(300)

tr

tr

td(Java) td(400)

\*//td[1]->SQL

\*(//td)[1]->SQL

\*(//td)[3]->Java

\*(//td[1])[2]->java

**IQ3)What is the difference between //a, //a[1], (//a)[1] 1.**all the links

1. all the first links
2. only first link

**IQ4)Derive the xpath which matches with last checkbox. Answer:** (//input[@type=’checkbox’])[last()]

**IQ5)Write a xpath to select first and last checkbox**

(//input[@type=’checkbox’])[1]|(//input[@type=’checkbox’])[last()]

|||ly

\*(//input[@type=’checkbox’]

\*(//input[@type=’checkbox’])[1]

\*(//input[@type=’checkbox’])[6]

**IMPORTANT LOCATORS:**

1. Id
2. name
3. linkText
4. xpath

**Note:**

In very few situations xpath written using a browser(Firefox)may not work in some other browser,in such cases we can use CssSelector.

**CONVERTING XPATH TO CSSSELECTOR:**

|  |  |
| --- | --- |
| **XPATH** | **CSSSELECTOR** |
| **1)** //button[@type=’submit’] | button[type=’submit’] |
| **2)** //input[@id=’UN’] | input#UN |
| **3)** //input[@class=’c1’] | input.c1 |
| **4)** //a | a |
| **5)** //tr/td | tr>td |
| **6)** //table//a | table a |
| **7)** //td/.. | Backward traversing is not supported in selenium |
| **8)** //td[text()=’Java’] | text() is not supported in selenium |

**IQ6) Can we use independent dependent concept in selenium Ans;**

No,because backward traversing is not supported in selenium

**Note: 1) get()** will enter the Url and wait till the page is completely loaded.Waiting time of get() is

**infinite.**

**2)findElement()** will search for specified element in the current page,if it is present it will returns the address of the element.If it is not there it will trow NoSuchElementFoundException IMMEDIATELY.

**SYNCHRONIZATION:**

Process of matching Selenium speed with application is called as synchronization .In order to synchronize the script we can use sleep() of Thread class as shown below;

Try

{

Thread.sleep(30000);

}

catch(InterruptedException e)

{

}

IMPLICITLY WAIT:

If we use sleep(),It will drastically increase maintenance consumes lot of time and space.Instead of this we can use implicitly wait statement of selenium.

**Example:** driver.manage().timeouts.implicitlyWait(30,TimeUnit.SECONDS);

In the above example 30SECONDS is used by all the findElement method and each findElement() waits upto 30 SECONDS.

After every half second it will search for element in page till the time out or till the element is located which ever comes earlier,This is called **POLLING PERIOD**.This is specified in a class called FluentWait.

The implicitelyWait() takes 2 arguments i.e,duration(long type) and Timeunit() such as DAYS,HOURS,**SECONDS**,**MINUTES**,MILLISECONDS,MICROSECONDS and NANOSECONDS

If element is not located even after the time out then FindElement() will throw NoSuchElementException

EXPLICIT WAIT:

\*If the method is other than findElement then to synchronize it we can use explicit wait.

\*webDriverWait itself is called as Explicitwait because we specify waiting condition Explicitly These conditions are available in ExpectedConditions class,these are also called as Predicates.

\*If specified condition is not specified even after the specified duration then explicitWait will throw TimeoutException(Selenium unchecked exception)

**Code:**

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.openqa.selenium.support.ui.ExpectedConditions;

**import** org.openqa.selenium.support.ui.WebDriverWait;

**public class** Demo010216 {

**public static void** main(String[] args)

{

//script to login to actitime

//open the browser

WebDriver driver=**new** FirefoxDriver();

//enter the url driver.get("<http://localhost/login.do>");

//enter the username driver.findElement(By.*id*("username")).sendKeys("admin");

//enter the password driver.findElement(By.*name*("pwd")).sendKeys("manager");

//click on login button driver.findElement(By.*xpath*("//div[text()='Login ']")).click();

//wait till logout link is visible within 30Sec

**WebDriverWait wait=new WebDriverWait(driver,30); wait.until(ExpectedConditions.*visibilityOfAllElementsLocatedBy*(By.*id*("logoutLink")));**

//get the title of home page and print it String title=driver.getTitle(); System.***out***.println(title);

}

}

**IQ7) What are the differences between ImplicitWait and ExplicitWait**

|  |  |
| --- | --- |
| **ImplicitWait** | **ExplicitWait** |
| 1) We do not specify any condition | We should specify the condition |
| 2) We can only handle findElement() and findElements() | We can handle any method |
| 3) After the timeout we get NoSuchElementException | After the timeout we get TimeOutException |
| 4) Time unit can be DAYS,HOURS,SECONDS etc | It can be only SECONDS |

**IQ8) Write a script to login and logout from the application without specifying any type of waiting duration(period)**

**Answer: import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.openqa.selenium.support.ui.ExpectedConditions;

**import** org.openqa.selenium.support.ui.WebDriverWait;

**public class** Demo010216 {

**public static void** main(String[] args)

{

//script to login to actitime

//open the browser

WebDriver driver=**new** FirefoxDriver();

//enter the url driver.get("<http://localhost/login.do>");

//enter the username driver.findElement(By.*id*("username")).sendKeys("admin");

//enter the password driver.findElement(By.*name*("pwd")).sendKeys("manager");

//click on login button driver.findElement(By.*xpath*("//div[text()='Login ']")).click(); **while**(**true**)

{

**try**

{

}

driver.findElement(By.*id*("logoutLink")).click();

**break**;

**catch**(Exception e)

{

}

}

}

**}**

**IQ9) How do you click on a button without using click()**

**Answer:** By pressing enter key

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.Keys;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public class** Demo01216 {

**public static void** main(String[] args)

{

WebDriver driver = **new** FirefoxDriver(); driver.get("https://demo.vtiger.com/");

driver.findElement(By.*xpath*("//button[text()='Sign in']")).sendKeys(Keys.***ENTER***); driver.findElement(By.*xpath*("//button[text()='Sign in']")).submit();

}}

\*We can also use submit() if button type=submit

<button type=”submit”>sign in</button>

\*We can also use java script to click on a button

**IQ10) How do you change the value present in the text box Ans:**

Using clear() and sendkeys()

WebElement un=driver.findelement(By.id(“username”)); Un.clear();

Un.sendkeys(“bhanu”);

**IQ11) How do you remove the value present in the text box without using clear() Ans:**

un.sendkeys(Keys.CONTROL+”a”+Keys.DELETE);

**IQ12)Write a script to copy paste the value present in one text box into another text box Ans:**

un.sendkeys(Keys.CONTROL+”ac”+Keys. CONTROL+”v”);

**IQ13)Write a script to print value present in the textbox.**

**Ans:** WebElement un=driver.findElement(By.id(“username”)); String v=un.getAttribute(“value”);

System.out.println(v)

**Limitation1:**

In Selenium we cannot store the password in encrypted format. **IQ14)How do you retrieve tooltiptext of an element Ans:** using getAttribute(“title”)

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.Keys;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public class** Demo01216 {

**public static void** main(String[] args)

{

WebDriver driver = **new** FirefoxDriver();

driver.get("https://demo.vtiger.com/");

WebElement chkBox=driver.findElement(By.*name*("remeber")); String tt = chkBox.getAttribute("title");

System.***out***.println("tt");

}}

**IQ15)Write a script to find the phone number of mubai in isrtc.com**

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.Keys;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public class** Demo01216 {

**public static void** main(String[] args)

{

WebDriver driver = **new** FirefoxDriver(); driver.get("https://demo.vtiger.com/");

WebElement chkBox=driver.findElement(By.*name*("remeber")); String tt = chkBox.getAttribute("title");

System.***out***.println("tt");

}

}

**IQ16)What is the difference between getAttribute() & getText()**

**Ans:** getAttribute() get the value of the specified attribute where as getText() is used to get the text of the specified element

**IQ17)Write a script to print x and y co-ordinates of an element.**

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.Point;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public class** Demo622016 {

**public static void** main(String[] args)

{

WebDriver driver = **new** FirefoxDriver();

driver.get("<http://localhost/login.do>");

Point p=driver.findElement(By.*id*("username")).getLocation();

**int** x=p.getX();

**int** y=p.getY(); System.***out***.println(x); System.***out***.println(y); driver.close();

}

}

**IQ18)Write a code to print height and width of the code**

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.Dimension; **import** org.openqa.selenium.Point; **import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public class** Demo622016 {

**public static void** main(String[] args)

{

WebDriver driver = **new** FirefoxDriver();

driver.get("[http://www.gmail.com"](http://www.gmail.com/));

Dimension d=driver.findElement(By.*id*("next")).getSize();

**int** h=d.getHeight(); **int** w=d.getWidth(); System.***out***.println(h); System.***out***.println(w); driver.close();

}

}

**IQ19)Write a code to print font size and color of the username text box in actiTime application**

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.Dimension; **import** org.openqa.selenium.Point; **import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public class** Demo622016 {

**public static void** main(String[] args) { WebDriver driver = **new** FirefoxDriver(); driver.get("<http://localhost/login.do>");

WebElement un = driver.findElement(By.*id*("username")); String fs = un.getCssValue("font-size"); System.***out***.println(fs);

String fc = un.getCssValue("color"); System.***out***.println(fc);

String ff = un.getCssValue("font-family"); System.***out***.println(ff);

driver.close();

}

}

OUTPUT:

14px Rgba(0,0,0,1)

MS Shell Dlg\32

**IQ20)Write a script to verify that login button is enabled**

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.Dimension; **import** org.openqa.selenium.Point; **import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public class** Demo622016 {

**public static void** main(String[] args) { WebDriver driver = **new** FirefoxDriver(); driver.get("<http://localhost/login.do>");

WebElement button = driver.findElement(By.*id*("loginButton"));

**if**(button.isEnabled())

{

}

**else**

{

}

System.***out***.println("Login Button is Enabled");

System.***out***.println("Login Button is not Enabled");

driver.close();

}

}

**IQ21)Write a script to verify that Next button in the gmail login page is visble(hint:using isDisplayed())**

**IQ22)Write a script to verify whether keep me logged in checkbox present in facebook login page is selected or not?(hint:isSelected())**

**Note:**

1. isSelected() can be also used on radio button
2. Importent mathods of WebElement Interface
   1. clear()
   2. **click()\***
   3. **getAttribute()\***
   4. getCssValue()
   5. getLocation()
   6. getSize()
   7. getTagName()
   8. **getText()\***
   9. isDisplayed()
   10. isEnabled()
   11. isSelected()
   12. **sendKeys()\***



* 1. submit()

##### last page:

InvalidStateElementException(Unchecked Selenium Exception)

## EXECUTING JAVA SCRIPT

Sometimes Selenium methods such as **click(),sendKeys() etc**.,may not work as an alternative option or work around we can use java script.

## EXECUTING JAVA SCRIPT MANUALLY:

**Step#1:**Open FireFox browser and open the required web page.press **F12** which opens firebug window.

**Step#2:**Click on console tab

**Step#3:**Type the javascript statement in the text box which is available at rite the java script statement in the text box which is available at the bottom of the firebug window and press enter

***Hi..!!***

**OK**

**>alert(‘hi’)**

## EXECUTING JAVA SCRIPT PROGRAMMATICALLY:

To run the java scripts programmatically in selenium we should use **excecuteScript()** of **JavascriptExcecutor**.Generally the object of the browser will be upcasted to WebDriver interface hence excecuteScript()will be hidden.In order to access this method either we should downcast it to RemoteWebDriver class or we should type cast it to JavascriptExcecuter interface

**I**

**I**

**Close();**

**WEB DRIVER**

**chromeDriver**

else{}

**executeScript(Str)**{}

**RemoteWebDriver Close() {} executeScript(Str){}**

**excecuteScript(Str)**

**JavascriptExecutor**

## C C

**C C**

**FirefoxDriver**

else{}

**Executescript(Str)**{}

**IQ23)Write a code to click() on the button using java script**

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.openqa.selenium.remote.RemoteWebDriver;

**public class** Demo062 {

**public static void** main(String[] args) {

Browser Constructor

Interface **WebDriver driver = new FirefoxDriver();** Up-Casting driver.get(“<http://localhost/login.do>”);

**RemoteWebDriver r=(RemoteWebDriver) driver;** Down casting r.executeScript("document.getElementById('loginButton').click()");

}}

**IQ24) Write a script to enter the text into text box without using sendKeys()**

**import** org.openqa.selenium.JavascriptExecutor;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.openqa.selenium.remote.RemoteWebDriver;

**public class** Demo062 {

**public static void** main(String[] args) { WebDriver driver = **new** FirefoxDriver(); driver.get("https://demo.vtiger.com/");



String c = "document.getElementById('username').value='abc'"; JavascriptExecutor j = (JavascriptExecutor) driver; j.executeScript(c);

}

}

**IQ25\*\*) How do you enter the text if the text box is disabled using JavaScript**

<html>

<body>

UN:<input id=”username” type=”text” disabled>

</body>

</html>

WebDriver driver = **new** FirefoxDriver(); driver.get("file:///c:/demo.html");

String c = "document.getElementById('username').value='bhanu'"; JavascriptExecutor j = (JavascriptExecutor) driver; j.executeScript(c);

**IQ26\*\*) Write a script to scroll to the bottom of the page**

WebDriver driver = **new** FirefoxDriver(); driver.get("<http://news.google.com/>");

String c = "window.scrollTo(0,document.body.scrollHeight)"; JavascriptExecutor j = (JavascriptExecutor) driver; j.executeScript(c);

|||ly String c = "window.scrollTo(0,document.body.scrollHeight/2)"; for half

String c = "window.scrollTo(document.body.scrollWidth,0)"; for complete right

**IQ27)Write a script to scroll to the element**

(**hint:** find the x and y co-ordinates of the element using getLocation(),pass them as argument for javaScript) Driver.findElement(By.id(“”)).getLocation()

Window.scrollTo(x,y)

**IQ28)Write a script to take the photo of a application import** java.io.File;

**import** java.io.IOException;

**import** org.apache.commons.io.FileUtils;

**import** org.openqa.selenium.OutputType;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.openqa.selenium.support.events.EventFiringWebDriver;

**public class** Demo062

{

**public static void** main(String[] args) **throws** IOException

{

WebDriver driver = **new** FirefoxDriver(); driver.get("[http://localhost/license.jsp"](http://localhost/license.jsp)); EventFiringWebDriver e=**new** EventFiringWebDriver(driver); File srcFile = e.getScreenshotAs(OutputType.***FILE***);

File destFile = **new** File("c:/abc.png"); FileUtils.*copyFile*(srcFile, destFile); driver.close();

}

}

**NOTE:**

**Limitation:**

\*using selenium we can take screen shot in PNG(portable network graphics) format only.we can not take the screenshot of popups,we cannot take screenshot of specific area in the page,we can not take the screenshot of multiple browser or desktop

\* If the page is very lengthy it will automatically takes the screenshot of complete page.

## ENCAPSULATION

“Process of hiding the data and binding with methods is called as encapsulation.”

Example:

public class A

{

private int i; public A()

{ i=10;

}

public int getValue()

{

return i;

}

}

Class B

{

Psvm()

{

A a1=new A(); S.o.p(a1.getValue());

}

}

For any given variable in java we should perform following steps;

1. Decleration
2. Initialization
3. Utilization

**NOTE:**

Usually initialize variables within constructor for encapsulation.

**Example:**

public class LoginPage

{

private WebElement unTextBox; public LoginPage(WebDriver driver)

{

unTextBox=driver.findElement(By.id(“username”));

}

public void setUserName()

{

unTextBox.sendKeys(“admin”);

}

}

**Example:**

**package qualitythought**;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**public class** LoginPage {

**private** WebElement unTextBox; **private** WebElement pwTextBox; **private** WebElement loginButton;

**public** LoginPage(WebDriver driver) {

unTextBox = driver.findElement(By.*id*("username")); pwTextBox = driver.findElement(By.*id*("pwd")); loginButton = driver.findElement(By.*id*("loginButton"));

}

**public void** login(String un, String pw) { unTextBox.sendKeys(un); pwTextBox.sendKeys(un); loginButton.sendKeys(un);

}

}

**package qualitythought**;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public class** Demo

{

**public static void** main(String[] args)

{

WebDriver driver=**new** FirefoxDriver();

driver.get("[http://localhost](http://localhost/)"); LoginPage l=**new** LoginPage(driver); l.login("admin","manager");

}

}

**Lastpage:**

StaleElementReferenceException(unchecked Selenium Exception) We get this when the page is reloaded

**package qualitythought**;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public class** Demo2 {

**public static void** main(String[] args) **throws** InterruptedException { WebDriver driver = **new** FirefoxDriver(); driver.get("[http://localhost](http://localhost/)");

LoginPage l = **new** LoginPage(driver); l.login("abc", "xyz"); Thread.*sleep*(3000);

l.login("admin", "manager");

}

}

### StaleElementReferenceException:

When we run the following code it will perform following steps;

**Step#1:** Open the browser

**Step#2:** Open the login page

**Step#3:** Create the object of Login page class by initializing its elements such as username (a1), password (b1), login (c1).

Where a1, b1, c1 are the address of the current address of the respective elements.

**Step#4:** Go to address a1 and type abc **Step#5:** Go to address b1 and type xyz **Step#6:** Go to address c1 and click

**Step#7:** Since invalid username and password are not entered after clicking on login button, it will display error message by reloading the complete page

**Step#8:** After reloading elements will be having different address i.e,username(x1),password(y1) and loginButton(z1)

**Step#9:** It waits for 3 seconds, it will try to enter “admin”in a1 and it is old address of username which does not exist anymore hence we get StaleElementReferenceException

## PAGE OBJECT MODEL



\*It is one of the java design pattern.

\*It is used to develop and test webpages.

\*In selenium we use page object model to avoid staleElementReferenceException and to improve the performance of the code.

\*In page object model we declare the element using FindBy Annotation(@FindBy) should be imported from following package;

Import org.openqa.selenium.support.FindBy; It has the following syntax; @FindBy(locator=”locatorValue”)

private webElement elementName;

We use initElement() [it is a static method] of pageFactory class to initialize all the elements of current class which are declared using @FindBy,it takes 2 arguments

1. WebDriver
2. Current object of the class

**IQ29)What happens if we do not use initElements()**

We get null pointer NullPointerException

**IQ30)How do you develop Pageobject model class without writing constructors**

In the POM class we include only declration and utilization and before calling any method of page object class in main method we use initElement statement.

**Example:**

@FindBy(id=”username”) Private WebElement unTextBox;

//no constructor

Public void setUserName(String un)

{

unTextBox.sendKeys(un);

}

**public static void** main(String[] args) **throws** InterruptedException { WebDriver driver = **new** FirefoxDriver(); driver.get("[http://localhost](http://localhost/)");



Log inPage l=new LoginPage(); pageFactory.initElements(driver,l); l.setUserName(“admin”);

We develop 2 types of classes

1. **POM class:** It is used to store the elements and its objects
2. **Test class:** It is used to execution purpose

**Note:**POM class is also called as Page object Repository because we use it to store elements(object of the page)

### TestNG: (Test Next Generation)

It is an unit testing framework.

It is a tool used by developers to run unit test cases.It can also be used by automation engineers to run multiple automation scripts and to generate the execution results.

TestNG is available as plugin for eclipse.To install it perform the following steps;

**#Step1:** Go to help in eclipse **#Step2:** Select eclipse marketplace **#Step3:** Search for TestNG

**#Step4:** Click install button of TestNG for eclipse

**#Step5:** Follow the default instructions till finish

After restarting the eclipse right click on java project Properties click on java build path Library tab click Add library Select TestNG Click next Click Finish click OK

##### TestNG class:



1. It is a java class which contains **Test method.**
2. Any method written using Test annotation(@Test).
3. When we run the TestNG class it automatically generates execution result in HTML format.The name of the file is ‘emailable-report.html’ which is present inside a folder called’test-output’
4. If the folder is not visible refresh the java project.

##### Example:

**package** pom;

**import** org.testng.Reporter;

**import** org.testng.annotations.Test;

**public class** Demo {

@Test

**public void** testA() {

Reporter.*log*("hi", **true**);

}

}

**Note:**

**\***To print the message in HTML report we use log reporter class to print same in console also we specify second argument as True

\*While developing TestNG class do not use **default package, main method and s.o.p**

### TestNG Suite

It is an xml file which contains list of TestNG classes which are to be executed,In order to create it right click on java project goto TestNG select convert to TestNG click Finish.It creates a file with the name testing.xml inside java project folder

##### Example:

<suite name=”Suite”parallel=”none”>

<test name=”Test”>

<classes>

<class name=”qsp.DemoA”/>

<class name=”qsp.DemoB”/>



</classes>

</test>

</suite>

##### Important tags of TestNG Suite:

* 1. <suite>
  2. <test>
  3. <classes>
  4. <class>

In order to execute it right click on the xml file goto runAs select TestNG Suite

**Note:** We can directly execute multiple classes by right clicking on the package or java project but it execute them in the random order

**IQ31)Can we have more than one Test method in TestNG class**

Yes

**IQ32)If multiple Test methods are present,In which order they will be executed?**

In the alphabetical order

**IQ33) How do you execute Test methods in required order**

Using ‘priority’

**Note:**

\*It always executes in ascending order.priority value need not be in continuous order it can be any integer values (+,-) and variables and decimals are not allowed.

\*The default priority value is “0”

\*If the priority is duplicate then those methods will be executed in alphabetical order

**IQ34) How do you write run a test method multiple times**

using invocationCount

**IQ35) What is the default invocationCount**

1

**Note:** \*If invocationCount is less than or equal to 0(<=0), it will not execute the test method.

\*For invocationCount we cannot use variables and decimal numbers.

### IMPORTANT ANNOTATIONS OF TestNG

1. **@Test**indicates Test method
2. **@BeforeMethod**indicates that the method should be executed before the execution ofevery @Test method
3. **@AfterMethod** This method is executed after every @Testmethod
4. **@BeforeClass** this method is executed at the beginning of the TestNGclass
5. **@BeforeClass** this method is executed at the ending of the TestNG class

**Example Program:**

**package** Demo;

**import** org.testng.Reporter;

**import** org.testng.annotations.AfterClass; **import** org.testng.annotations.AfterMethod; **import** org.testng.annotations.BeforeClass; **import** org.testng.annotations.BeforeMethod; **import** org.testng.annotations.Test;

**public class** Demo1 {

**public class** DemoA

{

@BeforeClass

**public void** openApp()

{

Reporter.*log*("open App",**true**);

}

@AfterClass

**public void** closeApp()

{

Reporter.*log*("close App",**true**);

}

@BeforeMethod

**public void** login()

{

Reporter.*log*("login",**true**);

}

@AfterMethod

**public void** logout()

{

Reporter.*log*("logout",**true**);

}

@Test(priority=1)

**public void** deleteuser()

{

Reporter.*log*("deleteuser",**true**);

}

@Test(priority=0,invocationCount=3)

**public void** edituser()



{

Reporter.*log*("edituser",**true**);

}

@Test(priority=-1)

**public void** registeruser()

{

Reporter.*log*("registeruser",**true**);

}

}

}

OUTPUT:

open App login registeruser logout

login edituser logout

login edituser logout

login edituser logout

login deleteuser logout

close App

**IQ36)How do you create dependency in TestNG**

Using “dependsOnMethods”

NOTE:

\*If both priority and dendency are specified it will consider the dependency

\*If both methods are independent then we will get TestNGException(unchecked TestNG exception)

**Example program: package** Demo;

**import** org.testng.Assert;

**import** org.testng.Reporter;



**import** org.testng.annotations.Test;

**public class** Demo2 {

@Test

**public void** createUser()

{

Reporter.*log*("createUser",**true**); Assert.*fail*();

}

@Test(dependsOnMethods={"createUser"})

**public void** deleteUser()

{

Reporter.*log*("deleteUser",**true**);

}

}

**OUTPUT:**

**Failed**: createUser

**Skipped**: deleteUser

**IQ37)How do you fail TestNG test**

using Assert.fail()

**IQ38)How do we compare actual value with expected value without using ifelse statement**

Using assertEquals() of Assert class

**Example:** Assert.assertEquals(actual,expected);

**IQ39)what are the methods available under Assert class**

1. asserEquals()
2. asserNotEquals()
3. assertTrue()
4. assertFalse()
5. assertNull() (used to check whether object is initialized or not)
6. assertNotNull()
7. assertSame()
8. assertNotSame()
9. fail()

Note:



\*All the above methods are static methods of Assert class.

\*\*\*If comparison fails then remaining statements of the current Test method will not be executed

\*To continue the execution even after the comparision fails we should use softAssert which has non- static methods.

**Example: package** Demo;

**import** org.testng.Reporter;

**import** org.testng.asserts.SoftAssert;

**public class** Demo3 {

{

@Test

**public void** create()

{

SoftAssert soft=**new** SoftAssert(); Reporter.log("Step1", **true**); soft.assertEquals("abc", "xyz"); Reporter.log("Step2", **true**); soft.assertAll(); Reporter.log("Step3", **true**);

}

}

}

**IQ40)What are the difference between Assert and SoftAssert**

|  |  |
| --- | --- |
| **Assert** | **SoftAssert** |
| If comparision fails,it will not execute the remaining statements of current method. | If comparision fails,it will execute the remaining statements of current method. |
| All the methods are static | All the methods are non-static |
| We do not call assertAll() | We should call assertAll() |

\*To verify major and critical features we use assert statements and to verify minor statements we use SoftAssert Statements

**IQ41) How do you rerun only failed TestNG classes?**

Using “testing-failed.xml” which is autogenerated by TestNG and it will be present inside test- output folder

**Important link:**

<http://testing.org/doc/documentation-main.html>

**AUTOMATION FRAMEWORK**

\*It is the standard guideline,best practices and rules which should be followed while automating the application.

\*We should use automation framework to have consistency.

\*In automation framework we have three stages:

1. Framework design
2. Framework Implimentation
3. Framework Excecution

# FRAMEWORK DESIGN

This is the initial stage where Automation lead or manager will specify the folder structures,naming conventions, types of files used etc., based on their past experience and project need.



**Example:**

Types of files used in the framework with their location

|  |  |
| --- | --- |
| **File type** | **Location** |
| .java | Javaproject/src |
| .class | Javaproject/bin |
| .html | Javaproject/test-output |
| .xml | Javaproject |
| .jar | Javaproject/jarfiles |
| .exe | Javaproject/exefiles |
| .xlsx | Javaproject/TestData |
| .bat |  |
| .war |  |

STEPS TO CREATE FOLDERS AND CONFIGURE THE FRAMEWORK:

**#Step1:** Goto required drive example:D drive and create a folder with the name BSSW5.

**#Step2:**I n eclipse goto file switch work space other.Browse and select newly created folder(D://BSSW5) click OK

**#Step3:** After restarting the eclipse create a java project with the name Automation

**#Step4:** Under the java project create a folder with the name jarfiles

**#Step5:** copy selenium jar file and paste it inside the above folder.

**#Step6:** Right click selenium.server.standalone file go to build path select add to build path.It will associate jar file with the java project



**#Step7:** Create a folder with the name exefiles under java project copy chromedriver.exe and iedriversever.exe files and paste the exefiles folder

**#Step8:** Create a folder with the name testdata inside javaproject folder,it will be used to store excel files

**#Step9:** Right click on java project goto properties click on java build path click on Add library under library tab select TestNG next finish ok

**#Step10:** Under src create 2 packages with a name pom and test scripts used to store pom class and testscripts of TestNG class

# FRAMEWORK IMPLIMENTATION

In this stage each automation engineer will convert the assigned test cases into automation scripts by developing 2 types of classes

* 1. POM class
  2. TestNG class First we should develop POM class

Automation team will select the test case for automation based on following 2 criterias;

1. It should be part of regression testing,this information will be provided by manual testing team
2. Test case shoud not have any manual intervensions;
   1. CAPTCHA(Completely Automated Public Turing test tell Computers and HumansApart)
   2. Bar Code Scanning
   3. Bio metric scanning, access cards,OTP,Credit cards etc., Because of the above reason 100% automation is not possible

#### SAMPLE TEST CASES

**TESTCASE 1: Valid Login**; **Precondition:** Login page should be present **Post Condition:** Application should be closed

**#step1:** Enter valid user name **#step2:** Enter valid password **#step3:** Click on login button

**#step4:** Click on logout button

**TESTCASE 2: Invalid Login:**

**#Step1:**Enter invalid username **#Step2:**Enter invalid password **#Step3:**click on login button

**#Step4:**Verify that error message is displayed

**TESTCASE 3: Verify build number:**

**#step1:** Login to the application using valid user name and password

**#step2:** click on help icon **#step3:** click on about actiTime **#step4:** verify build number **#step5:** click on close

**#step6:** click on logout

## STEPS TO DEVELOP A POM CLASS:

**#step1:** Number of POM class Should be same as number of web pages present on the application i.e.,for each web page there should be a POM class

**#step2:** Name of the POM class should be same as Title of the respective Web page ending with the word “Page”

**#step3:**I n each POM class we should declare the elements using @FindBy and initialized using PageFactory class

**#step4:** The action which should be performed by the elements should be developed of methods

**#step5:** First execute test cases manually which gives more clarity on the steps,which are to be automated

**#step6:** While executing Test cases note down title of the page,elements present on the page and actions which s ould be performed on the elements

Example:

PAGE 1:

**Title** Login

**Elements** username text box,password textbox,Login button,error message.

**Actions#step1:** Enter a value in the UN

**#step2:** Enter a value in password text field



**#step3:** click on Login button.

**#step4:** verify error message is displayed or not.

PAGE 2:

**Title** enter Time-Track

**Elements** Logout link,Help,about actiTime,Build number,close.

**Actions**

**#step1:** Click on logout link

**#step2:** click on help

**#step3:** click on about actiTime

**#step4:** click on close **#step5:** verify Build number **POM class for login Page:**

**package** pom;

**import** org.openqa.selenium.WebDriver; **import** org.openqa.selenium.WebElement; **import** org.openqa.selenium.support.FindBy;

**import** org.openqa.selenium.support.PageFactory;

**import** org.testng.Assert;

**public class** LoginPage {

@FindBy(id="username")

**private** WebElement unTextBox;

@FindBy(name="pwd")

**private** WebElement pwTextBox;

@FindBy(id="loginButton")

**private** WebElement loginButton;

@FindBy(xpath="//span[contains(text()='invalid')]")

**private** WebElement errMsg;

**public** LoginPage(WebDriver driver)

{

PageFactory.*initElements*(driver, **this**);

}

**public void** setUserName(String un)

{

unTextBox.sendKeys(un);

}

**public void** setPassword(String pw)

{

unTextBox.sendKeys(pw);

}

**public void** clickLoginButton()

{

loginButton.click();

}

**public void** verifyErrMsg()

{

Assert.*assertTrue*(errMsg.isDisplayed());

}

}

**POM class for Enter Time Track Page:**

**package** pom;

**import** org.openqa.selenium.WebDriver; **import** org.openqa.selenium.WebElement; **import** org.openqa.selenium.support.FindBy;

**import** org.openqa.selenium.support.PageFactory;

**import** org.testng.asserts.SoftAssert;

**public class** EnterTimeTrackPage { @FindBy(id = "logoutlink") **private** WebElement logoutLink;

@FindBy(xpath = "(//div[@class='popup\_menu\_arrow')")

**private** WebElement help;

@FindBy(linkText = "About actiTIME")

**private** WebElement aboutActiTime;

@FindBy(xpath = "//img[@title='close']")

**private** WebElement close;

@FindBy(xpath = "//span[contains(text(),'build')]")

**private** WebElement buildNumber;

**public** EnterTimeTrackPage(WebDriver driver) { PageFactory.*initElements*(driver, **this**);

}

**public void** clicklogoutlink() {

logoutLink.click();

}

**public void** clickhelp() {

help.click();

}

**public void** clickAboutActiTime() { aboutActiTime.click();

}

**public void** clickClose() {

close.click();

}

**public void** verifyBuildnumber(SoftAssert s, String eBuildNumber) { String aBuildNumber = buildNumber.getText(); s.assertEquals(aBuildNumber, eBuildNumber);

}

}

**DEVELOPING TestNG CLASS (automation script)**

1. For every manual TestCase we shpold develop TestNG class inside scriptspackage.
2. For all the Test cases there will be common steps such as preconditions and postconditions.instead of writing the code repetitively we use inheritance as shown below which increases codereusability;

# BaseTest Class:

**package** scripts;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.testng.annotations.AfterClass;

**import** org.testng.annotations.BeforeClass;

**public class** BaseTest {

**public** WebDriver driver;

@BeforeClass

**public void** preCondition() {

driver = **new** FirefoxDriver(); driver.get("[http://localhost](http://localhost/)");

driver.manage().timeouts().implicitlyWait(10, TimeUnit.***SECONDS***);

}

@AfterClass

**public void** postCondition() {

driver.close();

}

}

STEPS TO WRITE AUTOMATION SCRIPTS:

**#step1:** create a class under scripts package and the name of the class should be same as Respective Test case ID.



**#step2:** Extend it from BaseTest class

**#step3:** Create a Test Method and the name of the method should start with “test” and end with the class name

**#step4:** inside the test method write test case steps as in-line comments so that we will not skip any of the steps and it also helps the reviewer

**#step5:** After each in-line comment call the required method of POM class.

## TEST SCRIPT 1:

**package** scripts;

**import** org.testng.annotations.Test;

**import** pom.EnterTimeTrackPage;

**import** pom.LoginPage;

**public class** ValidLogin **extends** BaseTest { @Test

**public void** testValidLogin() {

// enter valid un

LoginPage l = **new** LoginPage(driver); l.setUserName("admin");

// enter the password l.setPassword("manager");

// click on login button l.clickLoginButton();

EnterTimeTrackPage e = **new** EnterTimeTrackPage(driver); e.clicklogoutlink();

}

}

#### HIDING METHODS OF OBJECT CLASS

In eclipse goto window preferences java appearance Type filters Add java.lang.Object OK OK

## TEST SCRIPT 2:

**package** scripts;

**import** org.testng.annotations.Test;

**import** pom.LoginPage;

**public class** InvalidLogin **extends** BaseTest

{

@Test

**public void** testInvalidLogin()

{

// enter invalid un

LoginPage lp = **new** LoginPage(driver); lp.setUserName("abc");

// enter invalid password lp.setPassword("xyz");

// click on login button lp.clickLoginButton(); lp.verifyErrMsg();

}

}

## TEST SCRIPT 3:

**package** scripts;

**import** org.testng.annotations.Test;

**import** org.testng.asserts.SoftAssert;

**import** pom.EnterTimeTrackPage;

**import** pom.LoginPage;

**public class** VerifyBuildNumber **extends** BaseTest { @Test

**public void** testverifyBuildNumber() { SoftAssert s = **new** SoftAssert();

// enter valid un

LoginPage lp = **new** LoginPage(driver); lp.setUserName("admin");

// enter valid password lp.setPassword("manager");

// click on login button lp.clickLoginButton();

// click on help

EnterTimeTrackPage e = **new** EnterTimeTrackPage(driver); e.clickAboutActiTime();

// verify Build Number e.verifyBuildnumber(s, "build 27261");

//click on close e.clickClose();



//click on logout e.clicklogoutlink(); s.assertAll();

}

}

**NOTE:** In order to open the login page and close the application only one time we can ude @BeforeSuite and @AfterSuite annotations.

# FRAMEWORK EXECUTION

To run all the scripts present in the frame work we use TestNG suite file.

**Example:**

Right click on the scripts package go to TestNG Select convert to TestNG Finish

Right click on testing.xml file select testing suite Refresh the java project which will display test- output folder open emailable-report.hyml file in the browser to see the execution purpose

<suite name=*"Suite"* parallel=*"none"*>

<test name=*"Test"*>

<classes>

<class name=*"scripts.ValidLogin"*/>

<class name=*"scripts.InvalidLogin"*/>

<class name=*"scripts.VerifyBuildNumber"*/>

</classes>

</test>

</suite> URL:https://poi.apache.org/download.html

**Section**:29 September 2015-POI 3.13 available

**SubSection:**Binary Distribution

**File**:poi-bin-3.13-20150929.zip

# DATA DRIVEN FRAMEWORK:

Testing the application with multiple inputs is called as data driven testing.If this feature is available in the framework then such type of file such as xl,xml,txt,csv.very frequently used is excel file we use the api provide by apache called POI.

**POI** (Poor obfuscation Implementation)

After downloading POI zip file extract it,It creates a folder with the name POI-3.13 within that we will have many files and sub folders only following 4 jar files are required;



1. Poi-3.13
2. Poi-ooxml-3.13
3. Poi-ooxml-schemes-3.13
4. Xml beans

Copy all the above 4 jar files paste them inside jar files folder of the framework.

Select all these 4 jar files using control right click, go to build path and select add to build path.

STEPS TO READ DATA FROM A CELL:

D: Book1.xlsx

0

2

1

0

1

1

0

2

Sheet 1

**#Step1:** Open xl fille(workbook)

**#Step2:** Goto sheet 1 **#Step3:** Goto row 0 **#Step4:** Goto cell 0 **#Step5:** Print cell

## NOTE:

1. Tostring method is present in Object class and it is inherited in all the java classes.
2. toString method will return address of the current object but in string class it is overrided instead of returning the address it returns the value of the String.
3. If we specify any reference variable as arguments for println method,it automatically calls toString().
4. In cell class also toString() is overrided,it returns value present in the cell,instead ofits address.
5. While reading data from xl sheet if specified sheet is not present or specified row or cell is blank we get NullPointerException.

#### \*\*\* CODE TO FETCH DATA FROM EXCEL:

##### Write a script to print content of the excel sheet.

**package** generics;

**import** java.io.FileInputStream;

**import** java.io.IOException;

**import** org.apache.poi.EncryptedDocumentException;

**import** org.apache.poi.openxml4j.exceptions.InvalidFormatException;

**import** org.apache.poi.ss.usermodel.Cell;

**import** org.apache.poi.ss.usermodel.Workbook; **import** org.apache.poi.ss.usermodel.WorkbookFactory; **public class** ExcelTest {

**public static void** main(String[] args) **throws** EncryptedDocumentException, InvalidFormatException, IOException

{

/\*FileInputStream fis = new FileInputStream("C:/book1.xlsx"); Workbook wb = WorkbookFactory.create(fis);

for(int i=0;i<2;i++)

{

for(int j=0;j<2;j++)

{

Cell c = wb.getSheet("Sheet1").getRow(i).getCell(j); System.out.println(c+" ");

}

System.out.println();

}\*/

}

}

##### Note:

String s = Excel.getCellvalue("C:/book1.xlsx", "Sheet1", 0, 0); System.***out***.println(s);

To count the number of rows present in the excel sheet we should use getLastRowNum() of sheet class it returns index of the last row(not the count)

To count the number of cells present in the row we should use getLastCellNum() of row

class,it returns the count instead of index

3 r.getLastCellNum

|  |  |  |  |
| --- | --- | --- | --- |
| A1  0 | B1  1 | C1  2 | 3 \* |
| A2  0 | 1\* | C2  2\* | D2  3\* |
| 0\* | 1\* | 2\* | 3\* |
| 0\* | 1\* | 2\* | D3  3 |

4

0

4

###### s.getLastRowNum

NullPointerException

**Assignment:** Write a script to print content of the excel sheet

# Generic Methods

If we can use the method in any project then such type of methods are called as generic methods,sometimes it is also called as utility library.

**Example:** Handling excel sheet methods ,handling database exception. We store these methods in separate package.

Example:

1. Create a package with the name generics under src folder then create a class withthe name Excel.
2. Develop static methods as shown below.

**package** generic;

**import** java.io.FileInputStream;

**import** org.apache.poi.ss.usermodel.Workbook;

**import** org.apache.poi.ss.usermodel.WorkbookFactory;

**public class** Excel {

**public static** String getCellvalue(String xlPath, String sheet, **int** row, **int** cell) { String v = "";

**try** {

FileInputStream fis = **new** FileInputStream(xlPath);

Workbook wb = WorkbookFactory.create(fis);

v = wb.getSheet(sheet).getRow(row).getCell(cell).toString();

} **catch** (Exception e) {

}

**return** v;

}

**public static int** getRowCount(String xlPath, String sheet) {

**int** rc = 0;

**try** {

FileInputStream fis = **new** FileInputStream(xlPath);

Workbook wb = WorkbookFactory.create(fis); rc = wb.getSheet(sheet).getLastRowNum();

} **catch** (Exception e) {

}

**return** rc;

}

}

**EXECUTING THE SCRIPTS BY TAKING DATA FROM EXCEL SHEET :**

1. Goto test data folder of the frame work
2. Create an excel file with the name DataInputs.
3. Rename a sheet as verify build number.
4. Enter the data as shown,save it and close it

|  |  |  |
| --- | --- | --- |
| **Username** | **Password** | **BuildNumber** |
| admin | manager | (build 27261) |

1. Update the test method present in the testNG class as shown below.

**Note:**

While specifying path of the excel file do not use absolute path,always use relative path. D:/BSSW5/Automation/testdata/Datainputs.xlsx

./testdata/DataInputs.xlsx

**Note:**

While entering test datas ensure that number of column is always fixed where as number of rows can be changed.

**File name:** data inputs.xlsx

**Sheet name:** invalid Login

|  |  |
| --- | --- |
| **Username** | **Password** |
| abc | xyz |
| admin | xyz |
| admin | bhanu |
| blank | blank |
| abc | manager |
| 123 | fg#$% |

**Assignment:**

Write a script to execute valid login script by taking multiple valid UN and PW from xl sheet

**IQ42)How do you pass a value from suite file into any method of TestNG class Ans:** Using Parameter

**Example:**

**#Step1:** Create a class as shown below;

**package** scripts;

**import** org.testng.Reporter;

**import** org.testng.annotations.BeforeClass;

**import** org.testng.xml.XmlTest;

**public class** ParameterTest {

@BeforeClass

**public void** preCondition(XmlTest xmlTest) { String v = xmlTest.getParameter("area"); Reporter.*log*(v, **true**);

}

@Test

**public void** testDemo(XmlTest x) {

String v = x.getParameter("city"); Reporter.*log*(v, **true**);

}

}

**#Step2:**Update TestNG.xml file as shown below

<suite name=*"Suite"* parallel=*"none"*>

<test name=*"Test"*>

<parameter name=*"city" value=”Bengaluru”*/>

<parameter name=*"area" value=”Basavanagudi”*/>

<classes>

<class name=”qsp.Demo”/>

</classes>

</test>

</suite>

**#Step3:** Excecute the suite file.

**Output:**

Basavanagudi Bengaluru **Note:**

**\***If we use xml test object in the testing class then it is mandatory to execute testNG classes from testNG.xml files.If we directly run the testNG classes then values of all the parameters will be null.

\*If we try to use those parameters to perform any action we get null pointer exception.

**\*\*\***When we execute testNG.xml file it automatically creates copy of this file(instance)and it will be automatically passed as arguments for the methods present in the TesstNG class.The data type should be always XmlTest



\*To retrieve the content of the xml,we use geters() of this class.

**IQ43\*\*)** How do you execute automation scripts on multiple browsers at the same time. By Using Parallel option available in testNG suite

**Example:** Step1: Update precondition method of BaseTest classas shown below; package scripts;

import java.util.concurrent.TimeUnit; import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver; import org.openqa.selenium.firefox.FirefoxDriver; import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass; import org.testng.xml.XmlTest;

public class BaseTest {

public WebDriver driver; @BeforeClass

public void preCondition(XmlTest xmlTest) {

String browser = xmlTest.getParameter("browser"); if (browser.equals("GC")) {

System.setProperty("webdriver.chrome.driver", "./exefiles/chromedriver.exe"); driver = new ChromeDriver();

} else {

driver = new FirefoxDriver();

}

driver.get(["http:](http://localhost/)/[/localhost](http://localhost/)"); driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

}

@AfterClass

public void postCondition() { driver.close();

}

}

**#Step3:** Update testNg.xml file as shown below;

<suite name=*"Suite"* parallel=*"tests"*>

<test name=*"TestGC"*>

<classes>

<class name=*"scripts.VerifyBuildNumber"*/>

<class name=*"scripts.InvalidLogin"*/>

<class name=*"scripts.ValidLogin"*/>

</classes>

</test>

<test name=*"TestFF"*>

<classes>

<class name=*"scripts.VerifyBuildNumber"*/>

<class name=*"scripts.InvalidLogin"*/>

<class name=*"scripts.ValidLogin"*/>

</classes>

</test>

</suite>

NOTE:

If we execute above TestNG.xml file,Since we specified parallel=”tests”it will create multiple threads.Each for one test block.In the above xml file we have two test blocks Hence it will create 2 threads.1st thread will execute all the scripts on the chrome browser and the 2nd thread will execute all the scripts on the FF browser parallel.This will reduce the total time taken to execute the scripts.

**IQ44)**How do you execute the framework from command prompt

**Answer:** Using following command;

Java –cp bin;jarfiles/\* org.testng.TestNG testng.xml Stepts to create batch file

**#Step1:**open the note pad and type the above command

**#Step2:**go to file and select save

**#Step3:**Navigate to the location where TestNG.xml file is present.



C:\BSSW5\Automation

**#Step4:** Specify the name as RunMe.bat

**#Step5:** To execute it double click on bat file

**\*\*\*Note:** To run the entire framework from command prompt we need jdk and TestNG. Eclipse IDE is not required Hence TestNG plugin is not required but we need following 4 jar files **1)**testng

**2)**jcommander.jar **3)**bsh **4)**snakeyaml

**How to get them?**

Ans: 1)The above jar files can be downloaded from the internet or we can copy from TestNG plugin i.e.,

Right click on Eclipse icon and select “open file location” Go to plugins folder Go to org.Testng folder go to lib folder and copy all the required 4 jar files and paste them inside jar files folder of the framework(totally 9 jar files should be there selenium1,poi4,testng4)

**SELENIUM GRID**

We test the application in different environments which are similar to production environment of the customer, this will be done during system compatibility testing. To run the framework on different computer we can copy paste the entire framework on to remote computer and execute it by double clicking on the batch file. In order to do this remote computer should have following softwares,

1. Build
2. Browsers: IE, FF, GC
3. JRE
4. Automation Framework Copy

If we need to test the application in multiple environment,

**Example:** In 20 different computers then we need to copy paste the frame work 20 different times and we need to execute the batch file 20 different times.

In order to execute the frame work on multiple remote computers, without copy pasting the frame work we use selenium grid

In order to execute the without copy pasting the frameworks we use Selenium grid. In selenium grid we will have 2 types of systems;

# HUB:

This is the computer where framework is present and it controlles the execution.In Selenium grid there will be only one hub.This system should have following softwares;

* 1. Build
  2. Browsers:IE,FF,GC
  3. JDK
  4. Automation FrameWork Copy
  5. Selenium jar files
  6. Eclipse IDE

# NODE:

It is a remote computer in which actual execution of the scripts takes place.We can have one or more nodes which will be communicated with the hub.We need following softwares in node;

* 1. Build
  2. Browsers:FF,GC
  3. JDK
  4. Selenium jar files
  5. chromedriver.exe

\*\*\*To implement this concept we should perform following steps;

1. start the Hub
2. start the node
3. Execute the framework using RemoteWebDriver
4. START THE HUB:
   1. Goto the computer where framework is present
   2. Open the command prompt and type the followingcommand;

Java –jar d:\s.jar-role hub

* 1. We should get a message “Selenium Grid hub is up and running”

**Note:** Default port number of hub is 4444 Change using:-port xxxx

1. START THE NODE:
   1. Go to remote computer and type the following command;

Java-jar e:\s.jar-role node-hub http://192.168.1.18:4444-Dwebdriver.chrome.driver=e:\cd.exe

(7 spaces)

* 1. It should display following message:

“The node is registered to the hub and ready to use”

**Note:** Port number of the node is 5555



1. EXECUTE THE FRAMEWORK USING REMOTE WEBDRIVER:
   1. Goto precondition metod of BaseTest class and update the code as below;

**Old code:**

driver = new ChromeDriver();

**New code:**

driver=**new** RemoteWebDriver(DesiredCapabilities.*chrome*());

* 1. Execute the framework by double clicking on batchfile.
  2. It will run all the scripts on node but result will be stored inside “test-ouput”folder of theframework.

**Note\*\*:** If hub or node is down we get UnreachableBrowserException(Unchecked Selenium Exception) add to last page.

# JENKINS

“Jenkins is a continuous integration tool used by developers to automatically create and install the build.”

Developers use Jenkins to perform following steps:

1. take the latest source code.
2. compile the code.
3. Run Unit scripts(WBT).
4. compress the compiled code(called Buid).
5. Install the build.
6. send e-mail notification.

**Note:** We can integrate framework with Jenkins so that it automatically execute the framework(batch file)after installing the build.this will avoid the manual intervension of running the framework manually by double clicking on the batch file.

**Download:** Jenkins-ci.org Jenkins.war

To implement the above concept we should perform following steps;

1)Install and configure Jenkins.(Developer) 2)Integrate framework with Jenkins.

3)Create the build.

1. **Install and configure Jenkins:**

This step will be done by development team

1. Download Jenkins.war file from Jenkins-CI.org website.
2. execute following command; java-jar d:\jenkins.war

It should display following message”Jenkins is fully up and running”

1. open the browser and enter Url of the Jenkins(localhost:8080)

It should display Home page of the Jenkins

1. Click on create new jobs
2. Specify the name
3. select first radio button”Free Style project.Click on OK and click save.
4. **Integrating FrameWork with Jenkins:**

This steps will be done by Automation engineer.

a)Open the home page of Jenkins— b)

1. \_click on advanced and select use custom workspace check box
2. Specify the location of the folder where the batch file is present
3. click on “add build “step
4. selectexecute “windows batch command”
5. Specify the name of batch file(RunMe.bat)
6. Click save
7. **Create Build:**
   1. This step will be done by the developers
8. open the Jenkins home page
9. click on name
10. click on”build now”

NOTE:When developer clicks on build now Jenkins will execute the framework after installing the build.

1. For every execution Jenkins will create a link under build history,So that we can view theexecution result.i.e.,Click on the link present under build history click on consoleoutput

**Automation flow diagram**

Interview questions on framework:

1. what is automation framework,what are thetypes/

Automation framework is standard rule best practice and guidelile which should be followed while automating the application testing.Based on the design we can categorise the framework into following types.

1. method driven automation framework
2. module driven automation framework
3. Data driven automation framework
4. Keyword driven automation framework
5. Hybrid automation framework

Note: All the above types are generic names.

* 1. The framework can be customized to the project need and it can be given with different names.such as Cucumber Framework
  2. Robot Framework
  3. Proptractor Framework etc.,

1. which framework you have used in your project andwhy?

We have used Hybrid framework which is the combination of )Data driven automation framework and method driven automation framework,We use methods to avoid the repeatation,we use multiple datas to test features thoroughly

1. Explain the flow diagrams of the framework
2. Explain the architecture of the framework
3. \*\*\*How many scripts you write per day. 2-5, it depends on the complexity
   1. How many scripts you execute per day

We execute all the scripts using framework automatically with the use of Jenkins. In a day it executes 150-200 scripts.

1. How many regression TCs you have in your project?

500

1. How many TCs are automated till today?

 90%

1. How many scripts you have written till today? 3\*200
2. what are your roles and responsibilities?



HANDLING MULTIPLE ELEMENTS:

Indexoutofboundsexception Add to last page(java uncked exception)

We use findelements()to handle multiple elements,which returns List<WebElement>

\*\*We should import the list from java.util package.we use following important methods of List;

1. size()
2. get(0)

It returns the size of the List(int)

It returns the element presnt in specified index(WebElement)

Example:Counting number of links present on flipkart.com and also printing text of all the links;

**import** java.util.List;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public class** Demo1 {

**public static void** main(String[] args) { WebDriver driver = **new** FirefoxDriver(); driver.get("[http://www.flipkart.com"](http://www.flipkart.com/));

List<WebElement> allLinks = driver.findElements(By.*xpath*("//a"));

**int** count = allLinks.size(); System.***out***.println(count); **for** (**int** i = 0; i < count; i++) {

WebElement link = allLinks.get(i); String text = link.getText(); System.***out***.println(text);

}

driver.close();

}

}

IQ)Write a script to print all the links exceptthe blank one If condition is used

Note:

For findelements we can use any of the 8 locators but generally we use xpath.

IQ)Write a script to select all the checkboxes from first to last and deselect all the checkboxes from last to first using both type of for loops.

 **import** java.util.List;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public class** CheckBox {

**public static void** main(String[] args) { WebDriver driver = **new** FirefoxDriver(); driver.get("file:///c:/demo.html");

String xp = "//input[@type='checkbox']";

List<WebElement> allCheckBox = driver.findElements(By.*xpath*(xp));

**int** count = allCheckBox.size();

**for** (WebElement checkbox : allCheckBox) { checkbox.click();

}

**for** (**int** i = count - 1; i >= 0; i--) {

WebElement checkbox = allCheckBox.get(i); checkbox.click();

}

}

}

IQ)How do you handle multiple elements in POM model?

In POM model we use @FindBy itself to handle multiple elements but we change the data type to List<WebElement>

**package** scripts;

**import** java.util.List;

**import** org.openqa.selenium.WebDriver; **import** org.openqa.selenium.WebElement; **import** org.openqa.selenium.support.FindBy;

**import** org.openqa.selenium.support.PageFactory;

**public class** DemoPage {

@FindBy(xpath = "//input[@type='checkbox']")

**private** WebElement checkBox; @FindBy(xpath = "//input[@type='checkbox']") **private** List<WebElement> allCheckBox;

**public** DemoPage(WebDriver driver) { PageFactory.*initElements*(driver, **this**);

}

**public void** ClickCheckbox() {

checkBox.click();

}

**public void** clickAllCheckbox() {

**for** (WebElement checkBox : allCheckBox) { checkBox.click();

}

}

}

WebDriver driver=**new** FirefoxDriver(); driver.get("file:///c:/demo.html"); DemoPage d=**new** Demopage(driver); d.clickAllCheckBox();

d.clickCheckBox();

IQ)What are the differences between findElement() and findElements()?

|  |  |
| --- | --- |
| findElement() | findElements() |
| It returns WebElement | It returns List<WebElement> |
| If the specified locator is matching with multiple elements,It returns first matching element. | If the locator matches with multiple elements it returns all matching elements |
| If the specified locator ids matching with non of  the elements,it will throw NoSuchElementException. | If the specified locator ids matching with none of the elements,it will return empty list |

IQ)How do you handle Autosuggestion Using findElements()

IQ)Write a script to search selenium in google and print all the auto suggestions and select one of the auto suggestions

**package** scripts;

**import** java.util.List;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public class** AutoSuggestions {

**public static void** main(String[] args) { WebDriver driver = **new** FirefoxDriver();

driver.manage().timeouts().implicitlyWait(10, TimeUnit.***SECONDS***); driver.get("[http://www.google.com](http://www.google.com/)"); driver.findElement(By.*id*("lst-ib")).sendKeys("selenium");

String xp = "//div[contains(text(),'selenium')]"; List<WebElement> allAST = driver.findElements(By.*xpath*(xp)); **int** count = allAST.size();

System.***out***.println(count);

**int** p = 0;

**for** (**int** i = 0; i < count; i++) {

String text = allAST.get(i).getText(); System.***out***.println(text);

**if** (text.equals("selenium webdriver")) { p = i;

}

}

allAST.get(p).click();

// write a code to handle if"web driver" is not present

}

}

IQ)Write a script to type letter’a’ in the from fied present in make my trip.com and print all the suggested options

HANDLING LIST BOX:

We use “Select” class of selenium to handle the list box.It should be imported from following package; Import org.openqa.selenium.support.ui.Select;

Example:

**package** scripts;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement; **import** org.openqa.selenium.firefox.FirefoxDriver; **import** org.openqa.selenium.support.ui.Select;

**public class** ListBox {

**public static void** main(String[] args) **throws** InterruptedException { WebDriver driver = **new** FirefoxDriver(); driver.manage().timeouts().implicitlyWait(10, TimeUnit.***SECONDS***); driver.get("[http://www.facebook.com"](http://www.facebook.com/));

WebElement listBox = driver.findElement(By.*id*("month")); Select select = **new** Select(listBox); select.selectByIndex(2);// select February Thread.*sleep*(1000); select.deselectByVisibleText("Mar");// select March Thread.*sleep*(1000);

select.selectByValue("4");

}

}

Important:

\*If Specified index text or value is invalid we get NoSuchElementException

\*We can use “Select” class itself to handle multi select list box.

\*If the specified option is duplicate in single select list box,It will select first matching option.

\*If te specified option is duplicate in multiselect list box,It will select all the matching options.

IQ)Write a script to count the number of options presnt in the list box and print all of them in reverse order.

 **package** scripts;

**import** java.util.List;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement; **import** org.openqa.selenium.firefox.FirefoxDriver; **import** org.openqa.selenium.support.ui.Select;

**public class** ListBoxReverse {

**public static void** main(String[] args) { WebDriver driver = **new** FirefoxDriver();

driver.manage().timeouts().implicitlyWait(10, TimeUnit.***SECONDS***); driver.get("file:///C:/Users/vidya\_s2/Desktop/ListBox.html");

WebElement listBox = driver.findElement(By.*id*("s1")); Select select = **new** Select(listBox);

List<WebElement> allOptions = select.getOptions(); **int** count = allOptions.size(); System.***out***.println(count);

**for** (**int** i = count - 1; i >= 0; i--) {

String text = allOptions.get(i).getText(); System.***out***.println(text);

}

driver.close();

}

}

IQ\*\*\*)Write a script to search for specified optionin the list box.

**int** found=0;

**for**(**int** i=0;i<count;i++)

{

aText=allOptions.get(i).getText();

**if**(aText.equals(eText))

{

found++;

}

}

System.out.println(found);

1. Not found
2. Found once 2,3,4,5… Duplicate Web Page:

<html>

<body>

<select id="s1" multiple>

<option value="i">idli</option>

<option value="v">Vada</option>

<option value="p">poori</option>

<option value="d">Dosa</option>

<option value="d">Vada</option>

</select>

</body>

</html>

**package** scripts;

**import** java.util.List;

**import** java.util.Scanner;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement; **import** org.openqa.selenium.firefox.FirefoxDriver; **import** org.openqa.selenium.support.ui.Select;

**public class** ListBox2 {

**public static void** main(String[] args) { System.***out***.println("option to Search?"); Scanner s = **new** Scanner(System.***in***); String eText = s.next(); System.***out***.println("Searching...");

**int** found = 0;

WebDriver driver = **new** FirefoxDriver(); driver.manage().timeouts().implicitlyWait(10, TimeUnit.***SECONDS***); driver.get("file:///C:/Users/vidya\_s2/Desktop/ListBox.html"); WebElement listBox = driver.findElement(By.*id*("s1"));

Select select = **new** Select(listBox); List<WebElement> allOptions = select.getOptions(); **int** count = allOptions.size();

**for** (**int** i = count - 1; i >= 0; i--) {

String aText = allOptions.get(i).getText();

**if** (aText.equals(eText)) {

found++;

}

**if** (found == 0)

{

System.***out***.println(eText + "Not found");

} **else if** (found == 1) {

System.***out***.println(eText + "found");

} **else** {

}

}

}

System.***out***.println(eText + "Duplicate");

}

IQ)Write a script to print content of list box in sorted order



**package** scripts;

**import** java.util.ArrayList; **import** java.util.Collections; **import** java.util.List;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement; **import** org.openqa.selenium.firefox.FirefoxDriver; **import** org.openqa.selenium.support.ui.Select;

**public class** ListBoxSorting {

**public static void** main(String[] args) { WebDriver driver = **new** FirefoxDriver();

driver.manage().timeouts().implicitlyWait(10, TimeUnit.***SECONDS***); driver.get("file:///C:/Users/vidya\_s2/Desktop/ListBox.html"); WebElement listBox = driver.findElement(By.*id*("s1"));

Select select = **new** Select(listBox); List<WebElement> allOptions = select.getOptions(); ArrayList<String> allText = **new** ArrayList<String>(); **for** (**int** i = 0; i < allOptions.size(); i++) {

String aText = allOptions.get(i).getText(); allText.add(aText);

}

Collections.*sort*(allText);

**for** (String text : allText) {

System.***out***.println("text");

}

}

}

IQ)Write a script to print content of the lis box in sorted order when th list box had numerical values. Example:Day list box present in FaceBook page

Hint Convert strint into number using integer.parsing and store them in array lest of integer then use sort method of collections:

**package** scripts;

**import** java.util.ArrayList; **import** java.util.Collections; **import** java.util.List;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement; **import** org.openqa.selenium.firefox.FirefoxDriver; **import** org.openqa.selenium.support.ui.Select;

**public class** NumericalSort {

**public static void** main(String[] args) { WebDriver driver = **new** FirefoxDriver();

driver.get("https://[www.facebook.com"](http://www.facebook.com/));

Select select = **new** Select(driver.findElement(By.*id*("day"))); List<WebElement> allOptions = select.getOptions(); ArrayList<Integer> allNumber = **new** ArrayList<Integer>();

// in Facebook,1st option id "DAY" so we arestarting loop from index

// 1(not 0)

**for** (**int** i = 1; i < allOptions.size(); i++) {

String aText = allOptions.get(i).getText(); allNumber.add(Integer.*parseInt*(aText));

}

Collections.*sort*(allNumber);

**for** (Integer i : allNumber) {

System.***out***.println("i");

}

}

}

Note: UnsupportedOperationException Java unchecked exception.(you may only de-select option of multi select)

\*The selectBy() is present in select class can be used on single select as well as multi select list box.

\*deselect() can be used only on multiselect list box.In order to check whether the list box is multi select or not we should use “isMultiple()” of select class.

Example:

**package** scripts;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.openqa.selenium.support.ui.Select;

**public class** isSelect {

**public static void** main(String[] args) { WebDriver driver = **new** FirefoxDriver();

driver.get("https://[www.facebook.com"](http://www.facebook.com/));

Select select = **new** Select(driver.findElement(By.*id*("day"))); select.selectByIndex(1);

select.selectByValue("d"); select.deselectByVisibleText("Poori"); **if** (select.isMultiple()) {

select.deselectAll(); select.deselectByIndex(1); select.deselectByValue("d"); select.deselectByVisibleText("Poori");

} **else** {

}

}

System.***out***.println("It is not multi select listBox");

}

IQ)Write a script to select and deselect the option in reverse order.

**package** scripts;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.openqa.selenium.support.ui.Select;

**public class** ReverseOrder {

**public static void** main(String[] args) { WebDriver driver = **new** FirefoxDriver(); driver.get("file:///c...");

Select select = **new** Select(driver.findElement(By.*id*("s1")));

**int** count = select.getOptions().size();

**for** (**int** i = count - 1; i >= 0; i--) { select.selectByIndex(i);

}

**for** (**int** i = count - 1; i >= 0; i--) { select.deselectByIndex(i);

}

}

}

UnexpectedTagNameExceptionSelenium unchecked

NOTE\*\*:If the list box is developed without using<select> html tag,we can not use select class.If we use it we get UnexpectedTagNameException.To handle such type of text box we can use sendkeys() or click() as shown below;

**package** scripts;

**import** org.openqa.selenium.By; **import** org.openqa.selenium.Keys; **import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement; **import** org.openqa.selenium.firefox.FirefoxDriver; **import** org.openqa.selenium.support.ui.Select;

**public class** RareCaseInSelect {

**public static void** main(String[] args) { WebDriver driver = **new** FirefoxDriver(); driver.get("<http://localhost/login.do>"); driver.findElement(By.*id*("username")).sendKeys("admin"); driver.findElement(By.*name*("pwd")).sendKeys("manager"); driver.findElement(By.*id*("loginButton")).click();

WebElement listBox = driver.findElement(By.*id*("ext-comp-1001")); listBox.clear(); listBox.sendKeys("Moss"); listBox.sendKeys(Keys.***ENTER***);

}

}

Assignment:Write a script to perform following steps

1. login to actiTime application.
2. click on task menu.
3. click on projects and customers check box
4. Select second radio button present in the list box
5. select a checkbox
6. click on close.

IQ)\*\* Write a script for the following steps:

1. goto following website<http://www.plus2net.com/php_tutorial/ajax_dd3.php>
2. select india in first list box
3. select rajastan in second list box
4. print content in last list box

IQ) How do handle drop down menu?

\*Dropdown menu is an element on which if we move the mouse pointer it will display list of options to handle it we use moveToElement() of “Actions” class

**package** scripts;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement; **import** org.openqa.selenium.firefox.FirefoxDriver; **import** org.openqa.selenium.interactions.Actions;

**public class** DropDownMenu {

**public static void** main(String[] args) { WebDriver driver = **new** FirefoxDriver(); driver.get("<http://www.actimind.com/>");

String xp = "//span[text()='About Company']"; WebElement menu = driver.findElement(By.*xpath*(xp)); Actions actions = **new** Actions(driver); actions.moveToElement(menu).perform(); driver.findElement(By.*linkText*("BasicFacts")).click();

}

}

Note:ElementNotVisibleException selenium unchecheked exception IQ)write a script to perform following steps;

1. goto istqb.in
2. navigate to”foundation registration corporate registeration online registeration IQ)write a script to move the mouse pointer

on about us menu pesent in istqb.in and print all the sub menus IQ)How do handle context menu

1. right clicking on any element is called context click
2. when we right click on any element we get a list of options called as context menu.

\*to right click on any element we use contextClick() of actions class and to select the required option present in the context menu we type the shortcut such as “t” for ne tab,”w” for new window etc., using sendkeys() of actions class.

**package** scripts;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.openqa.selenium.interactions.Actions;

**public class** ContextMenu {

**public static void** main(String[] args) { WebDriver driver = **new**

FirefoxDriver(); driver.get("<http://localhost/login.do>");

WebElement link = driver.findElement(By.*linkText*("Actimind Inc.")); Actions actions = **new**

Actions(driver); actions.contextClick(link).perform(); actions.sendKeys("w").perform();

}

}

IQ)How do perform drag and drop option in selenium?

**package** scripts;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement; **import** org.openqa.selenium.firefox.FirefoxDriver; **import** org.openqa.selenium.interactions.Actions;

**public class** DragAndDrop {

**public static void** main(String[] args)

{

like");

WebDriver driver=**new** FirefoxDriver();

driver.get("<http://www.dhtmlgoodies.com/submitted-scripts/i-google->

Actions actions=**new** Actions(driver); String xp1="//h1[text()='Block 1]";

String xp2="//h1[text()='Block 3]";

WebElement source=driver.findElement(By.*xpath*(xp1)); WebElement target=driver.findElement(By.*xpath*(xp2)); actions.dragAndDrop(source, target).perform();

}

}

HANDLING FRAMES:

1. Web Page present inside another WebPage is called as Embedded WebPage.
2. Developer uses “iframe” or”frameset” html tag to create the embedded webpage.
3. If the element is inside the frame,browser will display this frame option in the conext menu when we right click on that element.
4. Before performing any action on the element which are inside the frame,we should transfer the control Into the frame using following statement “driver.switchTo.frame(arg);”

Where argument can be index of the frame(int),id of the frame(string),element off the frame(WebElement).

1. In order to switch back to the page from the frame,we should use following statement; “driver.switchTo().defaultContent();”
2. To transfer the control from child frame to parent frame we should use following statement; “driver.switchTo().parentFrame();”
3. If the specified frame is not found we get”NoSuchFrameException”(Selenium Unchecked Exception)

Example:

DemoB.html:-

<html>

<body>

t2<input type="text" id="t1">

</body>

</html>

DemoA.html:-

<html>

<body>

t2<input type="text" id="t2">

</body>

</html>

**package** scripts;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public class** Frames {

**public static void** main(String[] args) { WebDriver driver=**new** FirefoxDriver(); driver.get("file:///D:/DemoA.html");

driver.findElement(By.*id*("t1")).sendKeys("abc"); driver.switchTo().frame("f1"); WebElement f=driver.findElement(By.*className*("c1")); driver.switchTo().frame(f); driver.findElement(By.*id*("t2")).sendKeys("xyz"); driver.switchTo().parentFrame(); driver.findElement(By.*id*("t1")).sendKeys("123");

}

}

IQ)Write a script to validate invalid login scenario for the following webpage https://[www.zoho.com/crm/lp/login.html](http://www.zoho.com/crm/lp/login.html)

Important:

A if new page is loaded or current page id refreshed control will be automatically transferred to main page;

IQoracle)What are the different ways to switch back to main page; 1)driver.switchTo().defaultContent().

1. )driver.switchTo().parentFrame().
2. driver.Navigate().refresh() POPUP HANDLING:

In selenium writing the code to handle the popup depends on type of the popup. With Respect to selenium we can catogarize the popups into following types;

1. Alert an confirmation popup
2. hidden division popup
3. file upload popup
4. file download popup
5. Child browser popup
6. Window popup Note:

NoAlertPresentException(Selenium Unchecked Exception)

UnhandledAlertException(Selenium Unchecked Exception) ALERT AND CONFIRMATION POPUPS:

Characterstics:

1. We can move the popup.
2. We cannot inspect the popup
3. If the popup has warning symbol with OK button(Trianle with !) it is alertpopup
4. If the popup has confirmation symbol(?) with OK and CANCEL button--.confirmation popup. NOTE:They are also called ass JavaScript popups.

To handle the alert and confirmation popups first we transfer the control using switch to alert statement a d then we use any of the following method of alert interface;

1. gettext() to get the message
2. accept To click OK button
3. dismiss() to click on CANCEL or CLOSE button

If popup is closed control will be automatically transferred back to main page.Whhile performing the action if the popup is not present we get NoAlertPresentException.

If we try to perform action on the page without closing the action we get UnhandledAlertException

**package** scripts;

**import** org.openqa.selenium.Alert; **import** org.openqa.selenium.By; **import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public class** PopupAlertAndConfirmation {

**public static void** main(String[] args) { WebDriver driver =

**new** FirefoxDriver(); driver.get("https://[www.isrtc.co.in/eticketing/loginHome.jsf](http://www.isrtc.co.in/eticketing/loginHome.jsf)"); driver.findElement(By.*id*("loginbutton")).click();

Alert alert =driver.switchTo().alert(); String msg = alert.getText(); System.***out***.println(msg); alert.accept();

alert.dismiss();

}

}

1. HIDDEN DIVISION POPUPS:

Characterstics:

1. We cannot move the popup.
2. We can inspect the popup.
3. Popup will be colorfull
4. Generally these types of popups are developed using <div> html tag and it will be hidden hence it is called as “Hidden Division popup”
5. we use findElement() to handle Hidden Division popup Calender popup is a type of Hidden Division popup. **package** scripts;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public class** HiddenDivisionPopup {

**public static void** main(String[] args) { WebDriver driver =

**new** FirefoxDriver();

driver.manage().timeouts().implicitlyWait(10, TimeUnit.***SECONDS***); driver.get("[http://www.yatra.com](http://www.yatra.com/)"); driver.findElement(By.*id*("BE\_flight\_depart\_date")).click(); driver.findElement(By.*id*("a\_2016\_3\_18")).click();

}

}

IQ)Write a script to performe following steps; 1)goto makemytrip.com2)click on departure date 3)select today’s date.

1. FILE UPLOAD POPUP;

Characterstics:

1. Clicking on browse button will diasplay a popup with the title file upload. 2)We can move the popup but we cannot inspect it.

Note:This popup is used to “select a file to be uploaded” SOLUTION:

To handle file upload popup we specify absolute path of the file as arguments for send keys method as shown below.

**package** scripts;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public class** FileUploadPopup {

**public static void** main(String[] args) { WebDriver driver = **new** FirefoxDriver(); driver.get("<http://www.2shared.com/>");

driver.findElement(By.*id*("upField")).sendKeys("D:\\Book1.xlsx");

}

}

1. Relative path is not supported
2. forward slash is not supported

To upload multiple files we shoud specify absolute path of each file using comma as a separator in sendkeys method.

If file upload popup is created without browse button like attatchment icon in Gmail.In such cases we cannot use sendKeys method.We should use third party tool such as AutoIt.

Limitations:

1. We cannot handle file upload popup of ot has attatchment icon instead of browse button.
2. FILE DOWNLOAD POPUP:

Characteristics;

1. We can move the popup
2. we cannot inspect the popup. 3)It will have 2 radio buttons: a)open with

b)save file

SOLUTION:To handle file download popup we use setPference() of FirefoxProfileclass

**package** scripts;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.openqa.selenium.firefox.FirefoxProfile;

**public class** FileDownloadPopup {

**public static void** main(String[] args) { FirefoxProfile profile = **new**

FirefoxProfile();

// if file is.zip then do not display popup download it directly String key = "browser.helperApp.neverAsk.saveToDisk";

String value = "application/zip"; profile.setPreference(key, value);

// open browser with above setting WebDriver driver = **new** FirefoxDriver();

driver.get("<http://docs.seleniumhq.org/download/>"); String xp = "//td[text()='Java'../td[4]/a"; driver.findElement(By.*xpath*(xp)).click();

}

}

Example:

1. Setting is called aspreference.
2. To change thesetting we use “setPreference()” of FirefoxProfile class.Thismethod takes arguments(key and value.To know the details about key and value follow below link;

“http://kb.mozillazine.org/About:config\_entries”

In Seelenium there is no option to handle file download popups in other browsers.Hence we use 3rd party tool such asAutoit

Limitation 6:

In Seelenium there is no option to handle file download popups in other browsers.Hence we use 3rd party tool such asAutoit

1. CHILD BROWSER POPUPS;

Characteristics:

1. We can move the popup 2)We can inspect the popup
2. We have minimiz,maximize options 4)It will have address bar

IQ)How do you handle child browser popup driver.switchTo().window(wh); IQ)What is window handle?

It is a “Unique alpa-numeric string of the Browser”

IQ)what is the difference between getWindowHandle() andgetWindowHandles()?

 getWindowHandle() It returns window handle of current browser(return type is String). getWindowHandles() it returns windowHandle of all browser.

IQ)What is the difference between close() and quit()?

close() This closes the current browser quit() close all browsers.

it will

IQ)write a script to count number of browsers opened by selenium IQ)write a script to print window handle of all the browsers.

IQ)Write a script to print title of all the browsers?

IQ)Write a script to clase all the browsers without using quit



**package** scripts;

**import** java.util.Set;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public class** ChildBrowserPopup {

**public static void** main(String[] args) { WebDriver driver =

**new** FirefoxDriver();

Set<String> allWH = driver.getWindowHandles(); System.***out***.println(allWH.size());

**for** (String v : allWH) { System.***out***.println(v); driver.switchTo().window(v); String title = driver.getTitle(); System.***out***.println(title);

driver.close();

}

}

}

IQ)Write a code to close specified browser. IQ)Write a script to close only parent browser Use driver.close()

IQ)Write a script to close only child browser;

Exceptions:

21)AWTException.-->Checked selenium Exception 22)NoSuchWindowException Selenium unchecked exception

**package** scripts;

**import** java.awt.Dimension; **import** java.awt.Rectangle; **import** java.awt.Robot; **import** java.awt.Toolkit;

**import** java.awt.image.BufferedImage;

**import** java.io.File;

**import** javax.imageio.ImageIO;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public class** ScrrenShot {

**public static void** getScreenShotofDesktop()

{

**try**

{

}

Robot r=**new** Robot();

//get the current size of the desktop

Dimension d = Toolkit.*getDefaultToolkit*().getScreenSize(); Rectangle screenRect = **new** Rectangle(d);

//take screen shot of complete desktop

BufferedImage img = r.createScreenCapture(screenRect);

//save to disk

ImageIO.*write*(img, "jpg", **new** File("d:/desktop.jpg"));

**catch**(Exception e)

{

}

}

**public static void** main(String[] args) **throws** InterruptedException

{

WebDriver driver = **new** FirefoxDriver(); driver.get("<http://docs.selenium.org/download/>"); driver.findElement(By.*linkText*("2.52.0")).click(); Thread.*sleep*(3000); *getScreenShotofDesktop*(); driver.close();

}

}

1. WINDOW POPUP:

Characteristics:

1. If the popup displayed in the application is not alert and confirmation,hidden division,file upload,file download or child browser then it is “Window popup” SOLUTION: In selenium there is no option to handle this popup

Limitation 7) In selenium there is no option to handle this window popup In order to handle window popup we use third party tools such as “Autoit”

AutoIt:

\*It is a free window based automation tool.

\*It can be downloaded from following URL; “<http://www.autoitscript.com/site/autoit/downloads/> File name: autoit-v3-setup.exe

1. double click on the set of file
2. follow the default instructions available in set of wizard(next,next..) 3)Click finish

Steps to inpect elements in auto it: 1)Goto start All ProgramsAutoit 2)Select Autoit window info

1. drag and drop finder tool on the requiredelement.

STEPS TO WRITE SCRIPT IN AUTOIT:

1. Goto all programs AutoIt select script editor 2)Write the code as shown below; WinWaitActivw(“Calculator”)

WinClose(“Calculatorr”)

1. goto filesave
2. navigate to the equired loacation
3. specify the name ex.script1.-->clicksave script.au3



it

1. goto tool and click complie

tools clickgo

will create an.exe file on the samelocation 7)to run the autoit tool goto

1. We can also execute it directly by double clicking on the .exe file

To run/execute the script from java,we can use “exec()” of Runtime class Runtime.getRuntime().exec(“C:\\Windows\\system32\\calc.exe”); Runtime.getRuntime().exec(“D;\\Script1.exe”);

IQ)Write a script to perform following steps; 1)open firefox browser

1. enter the url of selenium downloadpage
2. click on the link”2.52.0”
3. Activate the windowpopup
4. Press left arrow key so that control can be transferred from cancel button 6)press enter key so that it clicks on save file button.
5. AutoIt Script:

WinWaitActive(“Opening selenium-server-standalone-4.x.jar”) Sleep(1000) Send(“{LEFT}”)

Sleep(1000) Send(“{ENTER}”)

1. Save the above code ass script1 in D-drive which creates .exefile. 9)Write the above code in java and executeit;

**package** scripts;

**import** java.io.IOException;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public class** AutoIt {

**public static void** main(String[] args) **throws** IOException { WebDriver driver = **new** FirefoxDriver(); driver.get("<http://docs.selenium.org/download/>"); driver.findElement(By.*linkText*("2.52.0")).click(); Runtime.*getRuntime*().exec("d:\\script1.exe");

}

} MAVEN

:

Maven is a dependency tool with respect to selenium it is used to downloadlatest selenium jar file and run the automation framework.To implement this concept we should perform following steps;

1. Convert java project into maven project. 2)Specify the dependency.
2. Execute the framework using pom.xml(Project objectmodel).
3. Right click on the java project goto configure Select convert tomaven project Click finish.

It will generate a file with the name”pom.xml”( Project object model).Inside java project.Maven projects are indicated with the letter M above folder icon of the java project.

Specify the dependency:

\*Specify the jar file which are to be downloaded is called as dependency.

\*For each jar file we should specify following informations: 1)groupID

1. artifactID3)version.NOTE: Only for TestNG Scope of the dependency is test,For all the other it is compile(default option)

URL:”<http://docs.seleniumhq.org/download/maven.jsp>”

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>2.52.0<</version>

</dependency>

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<scope>test</scope>

<version>6.8<</version>

</dependency>

NOTE:We can directly copy paste above xml content into pom.xml file or goto dependency tab click on addspecify groupId,artifactId,version and scope click ok.

Executing framework using pom.xml:

\*After specifying the dependency,if we directly run pom.xml it can only download the jar file but it will not be knowing how to run the framework.

\*We have developed the framework using TestNG and we used testng.xml to executeit.

\*In order to run testng.xml from pom.xml we should use “sure fire”plugin which can be copied from following URL: <http://maven.apache.org/surefire/maven-surefire-plugin/examples/testng.html>

\*Copy the content from plugin to /plugin available in the above website.

\*goto pom.xml paste it just above </pugins> tag and save xml file.

\*Only for the first time right click on the pom.xml goto runAs and select Maven install second time on wards use maven test.

IMPORTANT NOTE:

If we run the framework directly using testng.xml; 1)It will not downloadd any jar files.

1. It will executes the scripts using the jar files which are listed under reference libraries.
2. After the execution result will be generated in test-ouputfolder. If we run the frame work using pom.xml then,
3. It will download the latest jar files from the internet into .m2 repository. 2)It executes the script using the jar files which are listed under maven dependencies folder.
4. After the execution the result will be generated under target folder.i.e, java project//target/surefire- reports/emailable-repert.html.
5. location of the .m2 repository will be “C:\Users\Administrator\.m2”
6. Using framework we cannot download other type of files such aschromdriver.exe

Specify the versions in maven:

|  |  |
| --- | --- |
| version | Meaning |
| [2.52.0] | Download only 2.52.0 version |
| [2.52.0,2.58.0] | Download only 2.52.0 or 2.58.0 in between versions are not allowed |
| [2.52.0,2.58.0) | Download the jar file with the version 2.52.0 to 2.58.0 |
| \*\*\*\*\* [2.52.0,) | Download 2.52.0 or any other above version |

\*We can also disable automatic update of the browser to avoid the failures.

JDBC

In order to connect to database and fetch the data ,so that we can verify it or use it.To use JDBC(Java Database Connection):

It will help us to connect us to data base.

\*Before writing JDBC code we should know following informations;

1. DataBase file ex.MySql
2. DataBase location ex:localhost or IP address
3. DataBase port numberex:3306
4. User name for the DataBase.:Ex:root
5. DataBase password:ex:Blank here
6. Name of the DataBase: ex: hello
7. Name of the table:ex:student
8. table colums:ex:Id,name,city
9. Sql query:ex: ex:select\*from student

In order to connect to database we should perform following steps;

1. Load the driver
2. Open the connection using connection string
3. execute sql statement
4. close the connection
5. we can download the driver for MySql database from the followingurl: https://dev.mysql.com/downloads/file/?id=13598

file: musql-connector-java-5.0.8-bin.jar

after downloading associate the above jar file to java project refer: <http://www.connectionstring.com/>

java code to open and close the database;

**package** scripts;

**import** java.sql.Connection; **import** java.sql.DriverManager; **import** java.sql.SQLException;

**public class** JDBC1 {

**public static void** main(String[] args) **throws** SQLException, ClassNotFoundException {

Class.*forName*("com.mysql.jdbc.Driver");

String cs = "jdbc:mysql://localhost:3306/<your sys name>"; String un = "root";

String pw = "";

Connection c = DriverManager.*getConnection*(cs, un, pw); c.close(); System.***out***.println("done");

}

}

IQ)Printing column header and table content of a database.

**package** scripts;

**import** java.sql.Connection; **import** java.sql.DriverManager; **import** java.sql.ResultSet; **import** java.sql.SQLException;

**public class** JDBC1 {

**public static void** main(String[] args) **throws** SQLException, ClassNotFoundException {

Class.*forName*("com.mysql.jdbc.Driver");

String cs = "jdbc:mysql://localhost:3306/<your server name> "; String un = "root";

String pw = "";

Connection c = DriverManager.*getConnection*(cs, un, pw); String sql = "select\*from student";

ResultSet r = c.createStatement().executeQuery(sql); **int** cc = r.getMetaData().getColumnCount(); System.***out***.println(cc);

**for** (**int** i = 1; i <= cc; i++) {

String cn = r.getMetaData().getColumnName(i); System.***out***.print(cn + " '");

}

System.***out***.println();

**while** (r.next()) {

**for** (**int** i = 1; i <= cc; i++) { String v = r.getString(i); System.***out***.println(v + " ");

}

System.***out***.println();

}

c.close();

System.***out***.println(" ");

}

} OUTPUT:

|  |  |  |
| --- | --- | --- |
| Id  1 | name  Akash | city  Agra |
| 2 | Bhanu | Bangalore |