**Protractor Interview Questions:**

1. **What is Protractor?**

* Protractor is an end-to-end test framework designed for Angular and Angularjs application.
* Protractor is node.js port of the webdriver.io which is the JavaScript implementation of selenium framework.
* Protractor is a Node.js program that supports test framework like jasmine Mocha and Cucumber.
* It works with both Angular and non-Angular JS applications equally well.

1. **What are the different types of locators in Selenium?**

Locator can be used as an address that identifies a web element uniquely within the webpage.

Angular Specific Locators

* by.binding
* by.exactBinding
* by.model
* by.repeater
* by.exactRepeater
* by.options

Other element locator’s introduced in Protractor:

* by.buttonText
* by.paritalButtonText
* by.cssContainingText
* by.deepCss

1. **What is Jasmine?**

* **Jasmine is an open source and BDD framework for testing JavaScript code.**
* It is a test framework, like JUnit (Java) and UnitTest (Python).
* It aims to run on any JavaScript-enabled platform. It doesn’t require a DOM and it has a clean, obvious syntax so that we can easily write tests.
* Jasmine supports BDD- s
* tyled tests using the describe- it syntax.

**Why unit Testing Framework (i.e Jasmine)?**

* + - 1. **Does not depend on any other JavaScript frameworks.**
      2. **Does not need any DOM**
      3. **Is very clean and obvious**
      4. **Open source**
      5. **Can manage/Control test scripts**

1. **Advantage of protractor**

* It has default waits which waits for Angular
* Easy to write an manage object
* ‘It supports BDD like Jasmine and mocha
* Image comparison is very easy
* Runs in multiple machines and supports parallel execution.

**Disadvantage:**

* Robot class is not there

1. **Difference between ElementFinder() and ElementfinderArray()?**

|  |  |
| --- | --- |
| **ElementFinder(** | **ElementfinderArray()** |
| represents a single element of an ElementArrayFinder | used for operations on an array of elements (as opposed to a single element). |
| can be treated as a WebElement for most purposes, in particular, you may perform actions (i.e. click, getText) | can call all(locator) and filter(filterFn) to return a new ElementArrayFinder modified by the conditions, and you can call get(index) to return a single ElementFinder at position ‘index’. |

1. What is Mocha?

* Mocha is a test runner
* Mocha is a JavaScript test framework for Node.js programs, featuring browser support, asynchronous testing, test coverage reports, and use of any assertion library.

1. **How to perform right click using web driver?**

Using Action class

Actions act=new Actions(driver);

WebElement c=driver.findElement((By.id(“”));

act.contextclick(c).perform();

1. **How to perform double click using web driver?**

Using Action class

Actions act=new Actions(driver);

WebElement c=driver.findElement((By.id(“”));

act.doubleclick(c).perform();

1. **How to perform Mouse Hover using web driver?**

Using Action class

Actions act=new Actions(driver);

WebElement c=driver.findElement((By.id(“”));

act.MovetoElement(c).perform();

1. **How to perform tooltip using web driver?**

Using Action class

//Create object of action class

Actions act=new Actions(driver);

WebElement c=driver.findElement((By.id(“”));

//To click on the element

act.MovetoElement(c).perform();

//title of the element and assign it to a string

String tooltipmsg=element.getAttribute(“title”);

//print the tooltip

System.Out.Println(“tooltip/Help message is”+tooltipmsg);

1. **Screenshot:**

File src= ((TakesScreenshot)driver).getScreenshotAs(OutputType.*FILE*);

**try** {

// now copy the screenshot to desired location using copyFile //method

org.apache.commons.io.FileUtils.*copyFile*(src, **new** File("D:/magic/Inlogic/error.png"));

}

**catch** (IOException e)

{

System.*out*.println(e.getMessage());

}

}

1. **Drag and Drop:**

Actions act=new Actions(driver);

WebElement Source=driver.findElement(By.id(“Sourcelement”));

WebElement Destination=driver.findElement(By.id(“destlement”));

act.dragandDrop(Source, Destination).build().perform();

1. **How to launch the browser?**

firefoxDriver driver=new firefoxDriver();

**To set the browser:**

Sytem.setproperty(“Webdriver.chrome.driver”,”D://Foldername//chromedriver.exe”);

chromeDriver driver=new chromeDriver();

Sytem.setproperty(“Webdriver.ie.driver”,”D://Foldername//iedriver.exe”);

InternetExplorerDriver driver=new InternetExplorerDriver();

1. ***What are the different types of navigation commands?***

|  |  |
| --- | --- |
| **Syntax** | **Description** |
| Navigation.back() | It takes back the user to the previous webpage |
| Navigation.forward() | Navigate to the next page |
| Navigation.refresh() | To refresh the current page |
| Navigation.to() | Launch a new browser window and navigate to specified URL |

1. **How can we handle web based pop up?**

**WebDriver** offers the users with a very efficient way to [handle these pop ups using Alert interface](http://www.softwaretestinghelp.com/handle-alerts-popups-selenium-webdriver-selenium-tutorial-16/).

|  |  |
| --- | --- |
| **Syntax** | **Description** |
| void dismiss() | The accept() method clicks on the “Cancel” button as soon as the pop up window appears |
| void accept() | The accept() method clicks on the “Ok” button as soon as the pop up window appears |
| String getText() | getText() method returns the text displayed on the alert box. |
| void sendKeys(String stringToSend) | sendKeys() method enters the specified string pattern into the alert box. |

**Syntax:**  
// accepting JavaScript alert  
                Alert alert = driver.switchTo().alert();  
 alert.accept();

1. **Dropdown:**
   * + 1. SelectByValue/ deselectByValue()
       2. SelectByVisibletext/ deselectByVisibleText()
       3. SelectByIndex/ deselectByIndex()

Select drpdown=new select(driver.findElement(By.id(“”));

Drpdwn.selectByValue(“green”);

Drpdwn.deselectByValue(“green”);

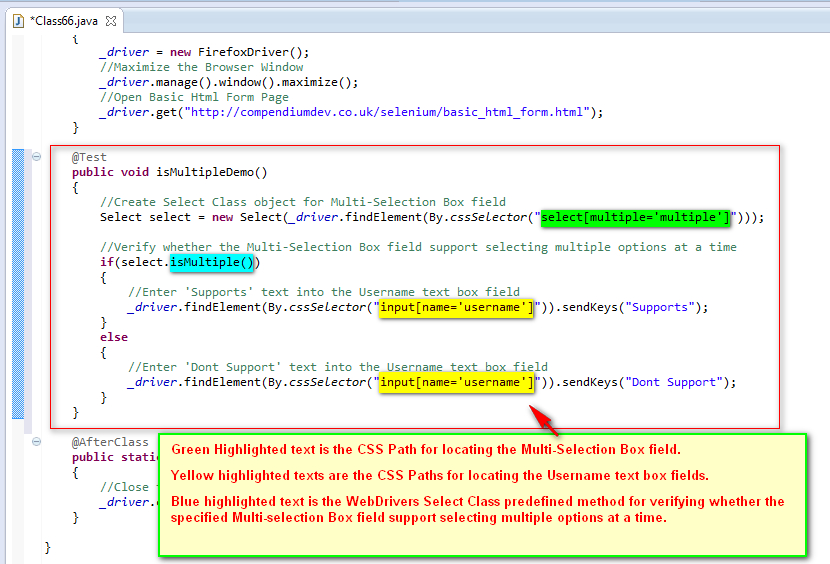
Drpdwn.selectByvisibletext(“lime”);

Drpdwn.deselectByvisibletext(“lime”);

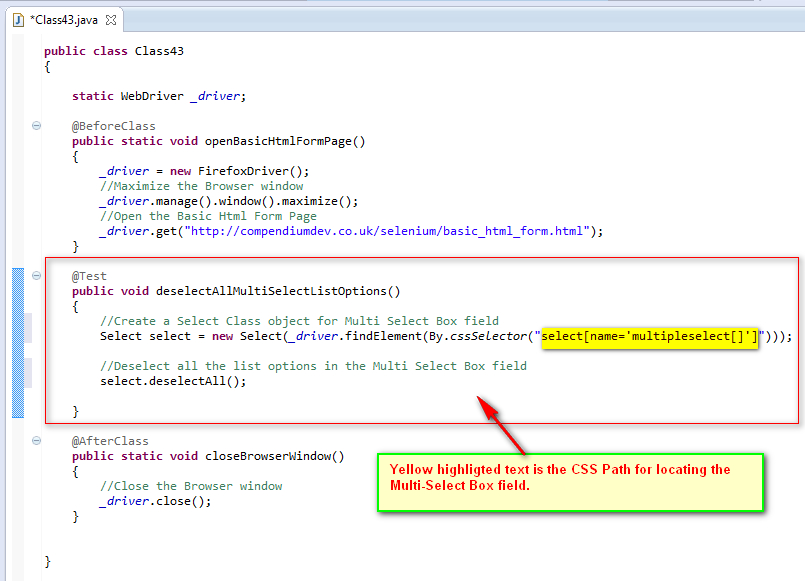
Drpdwn.selectByindex(2);

Drpdwn.deselectByindex(2);

**isMultiple()**



**deselectAll()**



1. **What is the main difference between the close() and quit() methods?**

|  |  |
| --- | --- |
| **close()** | **quit()** |
| Close the web browser window that the user is currently working on | Close all the opened web browsers |
| driver.close() | driver.quit() |

1. **Difference between Absolute and Relative xpath-CTS**

|  |  |
| --- | --- |
| **Absolute** | **Relative** |
| start selection from the document node/Start node | start selection matching anywhere in the document |
| Starts with / | Starts with // |
| e.g. “/html/body/p” matches all the paragraph elements | e.g. “//p” matches all the paragraph elements starts with p |
| It identifies the element very fast | it will take more time in identifying the element |

1. **Selenium IDE:**

* IDE stands for Integrated Development Environment
* Which is used for Record and play back the scripts

It is implemented as a **Chrome** and **Firefox** extension, and allows you to record, edit, and debug tests

* It can’t be used for mobile testing
* It does not support mouse, keyboard and Listener events
* It uses absolute [XPath to find web elements](http://www.inviul.com/writing-dynamic-xpath-selenium-webdriver/)

1. [**Selenium Grid**](http://www.softwaretestinghelp.com/selenium-grid-selenium-tutorial-29/)

* Selenium Grid is used for parallel testing or distributed testing. It allows us to execute test scripts parallelly on different machines

1. **Selenium RC**

* It does not offer Record & Play facility
* It is an API
* It supports multi-browser testing like Mozilla, Chrome, Safari, etc
* Selenium server required for Selenium RC
* It has easy to learn API
* It is not fully OOP based
* It uses absolute XPath to find web elements
* It cannot be used for mobile testing

1. **Selenium WebDriver:-**

* It is not enabled with Record & Play facility
* It is ready to work with all the browsers like Safari, Chrome, Mozilla, IE, etc.
* It supports all the languages like Java, Python, C#, PHP, Ruby, C++, etc.
* It does not require Selenium server
* It communicates in native language with browsers
* It supports mouse, keyboard and Listener events
* It is fully Object oriented
* It is ready for mobile testing
* It takes relative as well as absolute XPath

1. **How to exe parallel exe using selenium grid**

**Hub:**

* + - 1. Download latest selenium server file jar file
      2. Cmd>Type

java –jar selenium –server –standalone -3.3.1.jar –role hub

* + - 1. Hub ip address: http://192.168.1.164:4444

**Node:**

1. Download latest selenium server file jar file
2. Cmd>Type

java –jar selenium –server –standalone -3.3.1.jar –role node hub [http://12.168.1.164:4444/grid/register -port 5555](http://12.168.1.164:4444/grid/register%20-port%205555)

1. After exe the cmd ,Return to hub and hit the URL.

[http://192.168.1.164:4444- it](http://192.168.1.164:4444-%20%20it) will display the result

**In test script:**

Public class Grid\_SetUp {

Public static WebDriver driver;

Public static void main(String[] args) throws MalformedURLException, InterruptedException{

String URL = "http://www.DemoQA.com";

String Node = "http://192.168.1.164:4444/wd/hub";

DesiredCapabilities cap = DesiredCapabilities.firefox();

Cap.setBrowserName(“firefox”);

Cap.setplatform(“platform.xp”);

Driver=new RemoteWebDriver(new URL(nodeURL), capability);

driver.quit();

}

}

1. **Implicit wait:**

* It will tell to the web driver to wait for certain amount of time before throws “NosuchElementException”
* Default setting is 0

**Syntax: driver.manage().timeout().implicitlywait(10,TimeUnit.SECONDS);**

1. **Explicit wait:**

* It will tell to the web driver to wait for certain conditions(Expected conditions)or max time exceed before throwing “Element NotvisibleException”
* It will applied only for specified elements

**Syntax:** WebDriver Wait wait=new WebDriverWait(driver,20);

WebElement element=wait.until(ExpectedConditions.elementTobeclickable(By.Xpath(“”));

1. **Fluent wait:**

* It is used to tell the web driver to wait for a condition as well as frequency with which we want to check the condition before throwing an exception

**Syntax:** **Fluentwait wait=new FluentWait(driver,20)**

**.withTimeout(5000,SECONDS);**

**.PollingEvery(250, SECONDS);**

**.ignoring(NosuchException.class);**

**WebElement wait=wait.until(ExpectedConditions.elementTobeclickable(By.Xpath(“”));**

1. **Basic Excel Read:**

FileInputStream fis=new FileInputStream(“path of the file”);

//Create Workbook

Workbook wb=WorkbookFactory.create(fis);

//Create a sheet in a Excel

Sheet s=wb.getsheet(“sheetname”);

//Create row and column in Excel

String value=s.getRow(rownum).getcell(cellNum).getStringcellvalue();//Read data

FileOutputStream fos=new FileOutputStream(“path of the file”);//write data

Wb.write(fos);

1. **Database connection:**

Download the jar file of MySQL Connector

**Settings:**

* 1. Control panel>Administrative tool>click ODBC
  2. Go to User **DSN tab**>Click **Add** button
  3. Select database to connection
  4. Specify name of the DSN and click on save button

**Code:**

Class.forName(“Jdbc.odbc.Driver”);

s.o.pln((“Driver Loaded”);

String dbLocation=”c:\\..”;

Connection con=DriverManager.getconnection(“jdbc.odbc”:+dbLocation);

s.o.pln(“c/n created”);

Statement smt=con.createStatement();

Resultset rs=smt.executeQuery(“Select \* from table”);

While(rs.next())

{

String name=rs.getstring(“username”);

String password=rs.getstring(“password”);

String email=rs.getstring(“email”);

s.o.pln(“name is” +name+”password is “ +pass “emailed is” +email);

}

}

**28. What are multiple ways of refreshing the current browser?**

* Driver.navigate().refresh();
* SendKeys – Keys.F5
* Driver.get(driver.getCurrentURL)
* Driver.navigate().to(driver.getCurrentURL)

**29. How to**[**handle frame in WebDriver**](http://www.softwaretestinghelp.com/selenium-tutorial-18/)**?**

* <iframe></iframe> tag in HTML

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Method name** | **Description** | **Example** |
| 1 | switchTo.frame(int frameNumber) | Pass the frame index and driver will switch to that frame. | driver.switchTo().frame(0); |
| 2 | switchTo.frame(string frameNameOrId) | Pass the frame element Name or ID and driver will switch to that frame. | driver.switchTo().frame("iframe1"); driver.switchTo().frame("IF1"); |
| 3 | switchTo.frame(WebElement frameElement) | Pass the frame web element and driver will switch to that frame. | driver.switchTo().frame(iframeElement); |
| 4 | switchTo().defaultContent() | Switching back to Main page from Frame | driver.switchTo().defaultContent(); |

**30. Windows Handling**

Move the Window A (Parent) to Window B (Child) and close the child window then move back to parent window.

driver.getWindowHandle() :: is used to handle single window i.e. main window and return type is string  
driver.getWindowHandles() :: is used to handle multiple windows and return type is Set<string>

P.S.V.M()

{

System.setproperty(“Webdriver.geckodriver”,”D://geckodriver.exe”);

Webdriver driver=new FirefoxDriver();

driver.get(<http://wwwgoogle.com>);

//Store the current window handle

String parentwindow = driver.getWindowHandle();

//click some link that opens a new window

Driver.findElement(By,.xpath(“”)).click();

//Switch to new window opened

**for**(String childwind : driver.getWindowHandles())

{

driver.switchTo().window(childwind);

}

//Close the new window if that window no more required

driver.close();

//Switch back to original browser (first window)

driver.switchTo().window(ParentWindow);

1. **What are the limitations of Selenium?**

* Selenium supports testing of only web based applications
* Mobile applications cannot be tested using Selenium
* Captcha and Bar code readers cannot be tested using Selenium
* Reports can only be generated using third party tools like TestNG or Junit.
* As Selenium is a free tool, thus there is no ready vendor support though the user can find numerous helping communities.
* User is expected to possess prior programming language knowledge.

31. List the different types of drivers available in Selenium Web Driver

* Firefox driver
* Internet explorer driver
* Chrome driver
* Safari driver
* Opera driver
* Android Driver
* iPhone Driver
* HTMLUnit driver
* Remote web driver

**32. What are the advantages of Selenium?**

* It supports C#, PHP, Java, Perl, Phython
* It supports different OS like Windows, Linux and Mac OS
* It has got powerful methods to locate elements (Xpath, DOM , CSS)

**33. Tell me any webdriver common exceptions which you faced during test case execution.**  
**TimeoutException:** This exception will be thrown when command execution does not complete in given time.  
**NoSuchElementException**: WebDriver software testing tool will throw this exception when element could not be found on page of software web application.  
**NoAlertPresentException:** This exception will be generated when webdriver ties to switch to alert popup but there Is not any alert present on page.  
**ElementNotSelectableException:** It will be thrown when webdriver is trying to select unselectable element.  
**ElementNotVisibleException:** Thrown when webdriver is not able to Interact with element which is available in DOM but it is hidden.

|  |  |  |
| --- | --- | --- |
| **Exception** | **Meaning** | **Root cause** |
| ElementNotVisibleException | If the element In the page is not visible | HTML attribute set to ‘**hidden**’ |
| NoSuchElementException | When no element could be located from the locator provided. | Locators of the web element may have changed Elements are removed from the DOM Their coordinates have changed |
| NoAlertPresentException | When we try to switch to an alert but the targeted alert is not present. | Alert does not exist at particular session try{ driver.switchTo().alert().accept(); } catch (NoAlertPresentException E){ E.printStackTrace(); } |
| NoSuchFrameException | When we try to switch to a frame but the targetted frame is not present. | Invalid address of the frame Switchable frame does not exist |
| NoSuchWindowException | When we try to switch to a window but the targetted window is not present. | The possible root cause could be the invalid address of the window Switchable window does not exist |
| TimeoutException | Thrown when a command does not complete in enough time | WebDriver fails to perform activity in defined time |
| WebDriverException | When there is some issue with driver instance preventing it from getting launched. |  |
| Screenshot Exception | Impossible to capture screen |  |
| NosuchattributeExceptions | When the attribute of elemnt could not be found |  |
| StaleElementException | StaleElementException occurs if I find an element, the DOM gets updated then I try to interact with the element. | Sometimes it occurs due to slow server response |

**34. How can we get text of a web element in selenium?**

* By using gettext(),we can get the text of a webelement

**Syntax:** String ss=driver.findElement(By.xpath(“”)).gettext();

**35. How to type in a textbox using Selenium?**

* To enter the string in the textbox.

Ex: driver.findElement(By.id(“”)).sendkeys(“”);

**36**. **How can you find if an element in displayed on the screen?**

WebDriver facilitates the user with the following methods to check the visibility of the web elements. These web elements can be buttons, drop boxes, checkboxes, radio buttons, labels etc.

1. isDisplayed()
2. isSelected()
3. isEnabled()

**Syntax:**

**isDisplayed():**  
***boolean****buttonPresence = driver.findElement(By.id(“gbqfba”)).isDisplayed();*

**isSelected():**  
***boolean****buttonSelected = driver.findElement(By.id(“gbqfba”)).isDisplayed();*

**isEnabled():**  
***boolean****searchIconEnabled = driver.findElement(By.id(“gbqfb”)).isEnabled();*

**37. Radio box:**

**By using ID**

WebElement radioBtn = driver.findElement(By.id("toolsqa"));

radioBtn.click();

**38. Check box:**

**By using ID:**

WebElement chkbox=driver.findElement(By.id());

chkbox.click();

**39. getOptions**

Syntax: select.getOptions ();

Returns: List  
Purpose: Returns all the option elements displayed in this select tag (dropdown list)

**HTML Sample Code**

<**html**>

<**head**>

<**title**>Multi select Drop Down List Box</**title**>

</**head**>

<**body**> <**p**>What all devices do you listen to music on?</**p**>

<**select** name="Mobdevices" multiple="multiple">

<**option** value="0" selected> Please select</**option**>

<**option** value="1">iPhone</**option**>

<**option** value="2">Nokia</**option**>

<**option** value="3">Samsung</**option**>

<**option** value="4">HTC</**option**>

<**option** value="5">BlackBerry</**option**>

</**select**>

</**body**>

</**html**>

**Webdriver Code:**

*//To get all the options present in the dropdown*

List<WebElement> allOptions = se.getOptions();

**for** (WebElement webElement : allOptions)

{

System.**out**.println(webElement.getText());

}

**40. To get the Count of the total elements inside SELECT.**

Select oSelect = new Select(driver.findElement(By.id("yy\_date\_8")));

List <WebElement> elementCount = oSelect.getOptions();

System.out.println(elementCount.size());

**41. Print all the Options**

Select oSelect = new Select(driver.findElement(By.id("yy\_date\_8")));

List <WebElement> elementCount = oSelect.getOptions();

int iSize = elementCount.size();

for(int i =0; i<iSize ; i++){

String sValue = elementCount.get(i).getText();

System.out.println(sValue);

}

**42. Scroll**

**Syntax:**

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript(Script,Arguments);

**Ex:**

**Vertical**

JavascriptExecutor js = (JavascriptExecutor) driver;

// This will scroll down the page by 1000 pixel vertical

js.executeScript("window.scrollBy(0,1000)");

**Horizontal**

//This will scroll the page Horizontally till the element is found

js.executeScript("arguments[0].scrollIntoView();", Element);

**43. Scroll up and down**

jse.executeScript("window.scrollBy(0,250)", "");

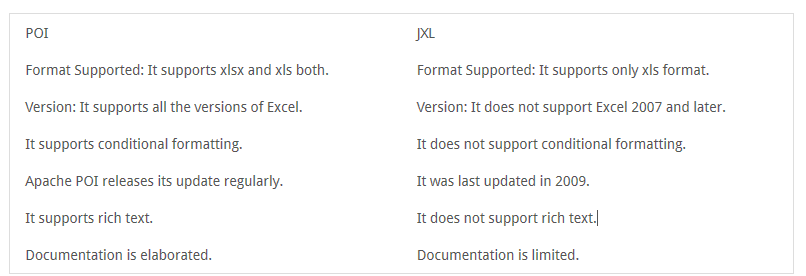
//Scroll vertically upward by 250 pixels

jse.executeScript("window.scrollBy(0,-250)", "");

**44. Roles & Responsibilities**

* Understanding the business requirements and preparing the FDS and test strategy
* Updating Framework functions when needed
* Developed and executing the Automation Test Scripts
* Reviewing the Test Reports and Preparing Test Summary Report.
* Preparation of test scenarios/cases/execution and test results.
* Report the defects / Observation on day to day basis
* Setup the environment for testing
* Attending the client calls and meetings
* Involving in giving daily and weekly status reports to clients.
* Involved in knowledge transfer to the newly recruited personnel.
* Implemented automation using Selenium WebDriver, JAVA, Selenium Grid, Maven.
* Extensively automated regression and functional test suites by developing over 237 test cases, 6 test suites using Selenium WebDriver, JAVA, JUnit.
* Implemented Page Objects framework, Hybrid framework and 21 Page classes from scratch to represent web pages.
* Developed Data Driven frameworks to retrieve test actions, test data from Excel files and SQL Databases.
* Configured Maven for JAVA automation projects and developed Maven project object model (POM).
* Used Maven, Selenium Grid to execute Selenium automation suites on different platform, browser combinations in parallel.
* Developed BDD tests using Cucumber by writing behaviors and step definitions. Developed required Selenium support code in JAVA for Cucumber.
* Wrote SQL queries extensively, queried database and generated test reports. Performed Purchase Orders Database testing by developing 14 SQL scripts.
* Performed Defect Tracking & Management in JIRA. Generated automated daily reports using JIRA API.
* Worked in a highly dynamic AGILE environment and participated in scrum and sprint meetings
* Assisted Manager by providing automation strategies, Selenium/Cucumber Automation and JIRA reports.
* Identified weaknesses in QA Processes, Web testing, Selenium Automation. Suggested & implemented improvements
* Building automation regression test suite for the application which can be executed during each sprint release
* Developing re-usable methods for repeating steps
* Automated highly transactional e-commerce web application using Selenium WebDriver.
* Implemented Page Objects, Data Driven, Keyword Driven, Hybrid automation frameworks using Selenium WebDriver, JAVA, JUnit.fr

**45. Difference between POI and JXL:**



**46. Tell Me Something About Yourself?” question:**

1. My name is sasireka,
2. I belongs to thanjavur.currently residing in madipakkam
3. I am graduate inB,.E Computer science and engineering from kings college of engg
4. I am carrying 6.3 years of experience in Software Testing Industry. First company is parina software there I worked as a software engg and my second company is lattice bridge infotech worked as a software engg Currently I am working in inlogic Pvt. Ltd.working as Sr. software tester.
5. If I talk about my family, In my family there are 6 members My father, mother, husband, Me,grandma and my little daughter.
6. I have couple of hobbies listening music.GARDENING.  
   That’s all about myself.

**47. Robot class upload a file:**

* Capture mouse and keyboard events
* Can upload the files
* Robot Class can easily be integrated with current automation framework (keyword, data-driven or hybrid)

1. **creating Object of robot class**
2. Import the package

**import** **java.awt.AWTException**;

**import** **java.awt.Robot**;

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Example** |
| * keyPress(); | This will press Escape key on keyboard. | robot.keyPress(KeyEvent.VK\_ESC); |
| * mousePress(); | This will press Left mouse button | robot.mousePress(InputEvent.BUTTON1\_MASK); |
| * mouseMove(); | This will move the mouse pointer to X and Y co-ordinates | robot.mouseMove(coordinates.getX(), coordinates.getY()); |
| keyRelease(); | This will release the CAPS\_LOCK key. | robot.keyRelease(KeyEvent.VK\_CAPS\_LOCK); |
| mouseRelease(); | This will release Left mouse button. | robot.mouseRelease(Inpuo[=tEvent.BUTTON1\_MASK); |
| KeyDown | To press down arrow key of Keyboard we use | **robot.keyPress(KeyEvent.VK\_DOWN)** |
| Tab | To press TAB key of keyboard | **robot.keyPress(KeyEvent.VK\_TAB)** |
| Enter | To press Enter key | **robot.keyPress(KeyEvent.VK\_ENTER** |

Ex:

This will click on Browse button

driver.findElement(By.id("uploadname1")).click();

Robot robot = new Robot();

Thread.sleep(1000);

robot.keyPress(KeyEvent.VK\_ENTER);

robot.keyRelease(KeyEvent.VK\_ENTER);

Thread.sleep(1000);

robot.keyPress(KeyEvent.VK\_CONTROL);

robot.keyPress(KeyEvent.VK\_V);

Thread.sleep(1000);

robot.keyRelease(KeyEvent.VK\_CONTROL);

robot.keyRelease(KeyEvent.VK\_V);

Thread.sleep(1000);

robot.keyPress(KeyEvent.VK\_ENTER);

robot.keyRelease(KeyEvent.VK\_ENTER);

/\* driver.close();

driver.quit();

\*/

}

}

**Using AUTO it:**

* Download Auto IT script software.
* After download you will get two setup file as shown in below screen, first is **AutoIt version 3** setup and second is **Scitautoit3** .
* For Installing AutoIT-Click on both AutoIT setup one by one .
* After successfully installation - open up AutoIT Editor.
* click on 'SciTE.exe' file, the AutoIT editor opens

WinWaitActive("File Upload") // enter the title of the pop up

Send("Path of the file to enter")   // enter the path of the file to upload

Send("{ENTER}") / press enter

**In coding:**

Runtime.getRuntime().exec("D:\AutoIt\AutoItTest.exe");

**48. Setspeed and sleep()**

**Thread.sleep():**

* Will stop current thread for the specific period of time

**Ex:** Thread.sleep(2000)

**Setspeed:**

Set execution speed for specific amount of time,for every selenium command.

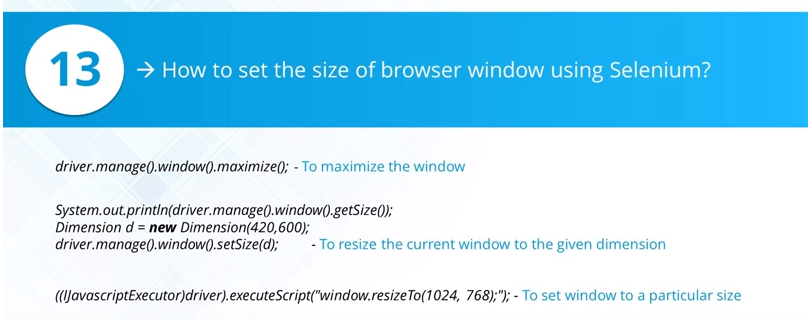
**Syntax:**

fSelenium.setSpeed(“2000”)

**49. How to retrieve css Properties of an element?**

Using get() method

Driver.findElement(By.id(“”))getCssValue(“Font-size”)



50**. Scroll Up and Down**

Actions a=**new** Actions(driver);

//scroll down

a.keyDown(Keys.*CONTROL*).sendKeys(Keys.*END*).build().perform();

Thread.*sleep*(1000);

a.keyDown(Keys.*CONTROL*).sendKeys(Keys.*HOME*).build().perform();

a.keyDown(Keys.*ALT*).sendKeys(Keys.*F1*).build().perform();

Some of the functions are

|  |  |  |
| --- | --- | --- |
| **S.No** | **Method name** | **Description** |
| 1 | Actions actions = new Actions(driver); actions.sendKeys(Keys.BACK\_SPACE).perform(); | Backspace |
| 2 | Actions actions = new Actions(driver); actions.keyDown(Keys.CONTROL).sendKeys(Keys.END).perform(); | Ctrl+End | Scroll to Bottom of the page |
| 3 | Actions action =new Actions(driver); action.sendKeys(Keys.TAB).sendKeys("keysToSend"); action.build().perform(); | TAB |
| 4 | driver.findElement(By.id("Value")).sendKeys(Keys.ARROW\_DOWN); | Down Arrow |
| 5 | driver.findElement(By.id("Value")).sendKeys(Keys.RETURN); or| driver.findElement(By.id("Value")).sendKeys(Keys.ENTER) | Enter/Return |

**51. Zoom in and Zoom out:**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Method name** | **Description** |
| 1 | WebElement html = driver.findElement(By.tagName("html")); html.sendKeys(Keys.chord(Keys.CONTROL, Keys.ADD)); | Zoom in |
| 2 | WebElement html = driver.findElement(By.tagName("html")); html.sendKeys(Keys.chord(Keys.CONTROL, Keys.SUBTRACT)); | Zoom out |
| 3 | WebElement html = driver.findElement(By.tagName("html")); html.sendKeys(Keys.chord(Keys.CONTROL, "0")); | Zoom 100% |

Some of the functions are:

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Method name** | **Description** | **Example** |
| 1 | moveToElement(toElement) moveToElement(WebElement) | **Parameter**s: toElement - Element to Move to **Purpose:** It moves the Mouse to the middle of the element or Mouse Hover | Actions act=new Actions(driver);  WebElement element = driver.findElement(By.id("ctl01\_MainContent\_ucPatientBanner\_lblPatientName")); act.moveToElement(element).perform(); |
| 2 | release() | **Purpose:** Release the current control |  |
| 3 | clickAndHold(WebElement) | **Purpose:** Clicks without releasing the current mouse location |  |
| 4 | contextClick(WebElement) | **Purpose:** Click on element and hold until next operation Once **right click** ,we have multiple option to select | Actions action = new Actions(driver).contextClick(element); action.build().perform(); |
| 5 | doubleClick() | Performs a double-click at the current mouse location. |  |
| 6 | sendKeys(KEYS.TAB) | For keyboard events |  |

Call object name (Drag and drop of webElement’s name)

Build()-Final method used so that all the listed actions will be compiled into a single step.

Perform()-Method when executing the Action object as we designed.

**52. What locators you have used the most?**

Finding elements by ID is usually going to be the fastest option, because at its root, it eventually calls down to document.getElementById(), which is optimized by many browsers.

“**sometime ‘XPath’ is not work on Internet Explorer** **but ‘Id’ will work on any browser**”

**cssSelector**is best for **Internet Explorer**

**53. Why you prefer id ?**

1. Because id is the unique identifier for the element and it does not change. This makes ‘Ids’ a very explicit and reliable way to locate elements on the page.
2. Also, all browsers also have highly efficient methods to get an object on the page using their ids. This makes id locators the fastest type of locator in Selenium.

**54. WebDriver is a class or interface?**

Web driver is an interface.

**55. Have you created any Framework?**

If you are a beginner: No, I didn’t get a chance to create a framework. I have used the framework which is already available.  
If you are an experienced tester: Yes, I have created a framework (Or) No, but I have involved in the creation of the framework.

**56. What kinds of test types are supported by Selenium?**  
1. Functional Testing  
2. Regression Testing  
3. Sanity Testing  
4. Smoke Testing  
5. Responsive Testing  
6. Cross Browser Testing  
7. UI testing (black box)  
8. Integration Testing are supported by Selenium.

**57**. **Examples of Abstract Class and Polymorphism in real life project using Selenium WebDriver?**

These concepts are commonly used in framework development.  Abstract class is used in defining a common super class while writing Page Object Model layer of the framework.  We usually create an abstract class named **BasePage**to have all common members for every page written in this class e.g **getPageTitle()**.

Then each Page class (**HomePage, LoginPage, DashboardPage** etc.) inherit from **BasePage**.  Sometimes one may need to change the behavior of methods implemented in superclass.  So subclass has freedom to override that method where we use polymorphism.  This is how we use Abstract class in real projects.

Just to cite an example in Selenium itself;  By class is an abstract class.

Polymorphism is of types – Overloading and Overriding.

One example where Overloading can be used is in Logger design.  A logger class provides functions like

log.logInfo(String msg);

log.logInfo(String formatString, …)

These two functions are overloads of each other.  Overriding is typically used with inheritance (example given above with PageObjectModel)

**58. What is Object Repository?**An object repository is an essential entity in any UI automations which allows a tester to store all object that will be used in the scripts in one or more centralized locations rather than scattered all over the test scripts.

**59 . How can you handle network latency using Selenium?**  
You can use driver.manage().timeouts().pageLoadTimeout(); for network latency.

-It can execute bash scripts, shell scripts, ANT and Maven Targets.  
-It can be used to Publish results and send email notifications.

**60. Maven?**

**General Phrases used in Maven:**

* **groupId**: Generally groupId refers to domain id. For best practices company name is used as groupId. It identifies the project uniquely.
* **artifactId**: It is basically the name of the Jar without version.
* **version**: This tag is used to create a version of the project.
* **Local repository**: Maven downloads all the required dependencies and stores in the local repository called m2. More details regarding the same would be shared in the next topic.

### ****Build Life Cycle:****

Basic maven phases are used as below.

* **clean**: deletes all artifacts and targets which are created already.
* **compile**: used to compile the source code of the project.
* **test**: test the compiled code and these tests do not require to be packaged or deployed.
* **package**: package is used to convert your project into a jar or war etc.
* **install**: install the package into the local repository for use of another project.

**61.How many test cases you have automated per day?**  
It always depends on the application and test cases. But on and average we can automate 5 to 6 test cases per day. Ex: 1. For analyzing the test cases 2. Developing the script 3. Debugging and executing the script 4. Stabilizing the script

**62.What Is Pom in Maven?**  
POM stands for Project Object Model. It is fundamental Unit of Work in Maven. It is an XML file. It always resides in the base directory of the project as pom.xml. It contains information about the project and various configuration details used by Maven to build the project(s).

**63.What is Jenkins?**  
Jenkins is an open source continuous integration tool written in Java. It keeps a track on version control system and to initiate and monitor a build system if changes occur.

<https://seleniumautomationtester.wordpress.com/2018/02/01/interview-preparation-automation-testing-profile/>

**64.GIT:**

Git is a Distributed Version Control system (DVCS). It can track changes to a file and allows you to revert back to any particular change.

**65. Finding & Printing number of rows**

List<WebElement>rows=driver.findElements(By.tagName(“tr”));

System.out.println(“Number of Rows including headings:”+ rows.size());

**66. Get the number of columns.-<th>(column heading only)**

List<WebElement>columns = rows.get(0).driver.findElements(By.tagName(“th”));  
System.out.println(“Number of columns:”+ columns.size());

**we will print the headings**

Finding & Printing the column headings

for(int cnum=0;cnum<columns.size();cnum++)  
{  
System.out.println(columns.get(cnum).getText());  
}

**67.What is the priority in which we have to use the locators ?**  
 **Answer:**We have to use the locators in the below specified priority:

* First Priority : id locator
* Second Priority : name locator
* Third Priority: Class Name locator
* Fourth Priority: link text locator
* Fifth Priority: CSS Selectors locator
* Sixth Priority: XPath locator
* Seventh Priority: DOM locator

id, name, class name, link text, css selectors , xpath expressions and DOM are the different types of locators which we use in Selenium for locating the GUI elements on the web pages

**68. Find row and column count;**

Rows: List<WebElement> rows=htmltable.findElements(By.tagName(“tr”));  
Col:   List<WebElement> columns=rows.get(rnum).findElements(By.tagName(“th”))

**69. How to find out all the links in web page:**

List<WebElement> allLinks = driver.findElements(By.tagName(“a”));

**Selenium Grid:** It transparently distributes tests into Remote machines. That is, running multiple tests at the same time against different machines running different browsers and operating systems.

**70. Find font color of text:**

By using getCssvalue()  
For example:  
driver.findElement(By.id(“text”)).getCssValue(“color”);

**71.How to find Dynamic Web Elements?**

We can Dynamic web elements with the help of XPath or CSS.  
For Example:  
By using XPath:  
driver.findElement(By.xpath(“//div[contains(@id,’yui\_’)]”));  
By using CSS:  
driver.findElement(By.cssSelector(“div[id\*=’yui\_’)]”));

**72.What is the difference between getText() and getAttribute()**

**getText():** It is used to retrieve the text of a Text Element from the web page  
For Example:  
String Text = driver.findElement(By.id(“Text”)).getText();  
**getAttribute():** It is used to retrieve the text from a Text Field that is already eneterd into the text field.  
For Example:  
String Text = driver.findElement(By.id(“username”)).getAttribute();

**73. Enter:**

driver.findElement(By.id(“email”)).sendKeys(Keys.ENTER);

**74. Get width and Height:**

driver.findElement(By.xpath(“xpath of textbox ”)).getSize().getWidth();  
driver.findElement(By.xpath(“xpath of textbox ”)).getSize().getHeight();

**75. What is the difference between driver.get() and driver.navigate.to(“url”)?**

**driver.get():** To open an URL and it will wait till the whole page gets loaded  
**driver.navigate.to():** To navigate to an URL and It will not wait till the whole page gets loaded

**76. Why do you prefer selenium tool for automation?**

a. Free and open source  
b. Cross-Browser testing  
c. Platform compatibility  
d. supports multiple programming languages

**77. How to clear the text in the text box using Selenium WebDriver?**

WebDriver driver = new FirefoxDriver();

driver.get("https://www.gmail.com");

driver.findElement(By.xpath("xpath\_of\_element1")).sendKeys("Software Testing Material Website");

driver.findElement(By.xpath("xpath\_of\_element1")).clear();

78. About Project

* 1. **About Current project:**

**Project Title**: ManageMyHealth India

**Front end** is Net application and **Back end** : SQL Server2016

**Project description*:***

This is a Health care project

* + - This portal is for communication between Doctor and patients
    - We have separate login for Doctors and patient
    - Doctor can create the slots and book the appointment ,put clinical notes and add the health indication for patients
    - Patient can also book their appointments, and view their clinical notes
    - They can communicate via emails
    - Videos call facility is available for dr and patients
    - Doctor can make the video call to patient when the patient can login in the application in Tablets
    - We have integrate the medical device like BP, Stethoscope, Nonin
    - Once the reading taken it will pass through the blue tooth and will get displayed in Doctors video call screening
    - We are supporting windows application Doctor to Doctor communication via video call and device integration
    - For payments we are using smart card application
    - It’s a smart card we can using the patient UHID and write the details on the card

And will read the card using the smart card reader

* + - Doctor can manage the patient reports
    - Currently we are supporting the project for organisation by taking the Assessment ,Report and maintaining the lab result
    - To know their health risk by using our application

**Ultrasis:**

**Project Title**: Ultrasis

**Front end** is php and **Back end** : SQL Server2016

**Project description*:***

* + - It’s a online depression treatment program for the users
    - Register user will take the assessment and provide the report based on the assessment they can know their health risk
    - They can set their goals
    - It will provide some online programs how to deal with stress and depression
    - User can improve their health and well being
    - User can able to login via by using the App

**Vodafone**:

**Project Title**: Vodafone

**Front end** is .net and **Back end** : My SQL

**Project description*:***

* + - It’s a customer care project
    - By using IVR ,we are inputting manual and ASR through mobiles
    - It will prompting the Account summary, Data pack, SMS pack, Online recharge
    - It will support INB and OBD calls

**FMS:**

**Project Title**: FMS

**Front end** is .PHP and **Back end** : My SQL

**Project description*:***

* It’s a online application provide the solution for lawyers
* They can easily find their client and clients can easily find their department wise lawyers
* They can book the appointment online
* Maintain their files case related documents and evidence videos

79.To fetch the Window handling more than 2 windows:

System.setProperty("webdriver.chrome.driver", "C:\\teachmeselenium\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.get("<http://www.teachmeselenium.com/automation-practice/>");

//Storing current window's handle

String strFirstWindowHandle = driver.getWindowHandle();

//Clicking on a link that opens a new window

driver.findElement(By.linkText("Click Me to open New Window")).click();

//Storing the collection of all opened windows (in our case it would be 2)

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

//Iterating over all windows handles

for(String strWindowHandle: setWindowHandles){

                //If the window handle is not same as the one stored before opening up second window, it is the new window

                if(!strWindowHandle.equals(strFirstWindowHandle)){

                                //Switch to the new window

                                driver.switchTo().window(strWindowHandle);

                                //Print window title

                                System.out.println(driver.getTitle());

                                //Exit from loop

                                break;

                }

}

//Storing current window's handle. In our case it would be second window.

String strSecondWindowHandle = driver.getWindowHandle();

//Select the item "Second Window" from drop-down list

Select lstSecondWindow = new Select(driver.findElement(By.id("list-second-window")));

lstSecondWindow.selectByVisibleText("Second Window");

//Click on link that opens third window

driver.findElement(By.linkText("Click Me To Open Third Window")).click();

//Storing the collection of all opened windows (in our case it would be 3)

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

//Iterating over all windows handles

for(String strWindowHandle: setNewWindowHandles){

                //If the window handle is not same as the First and Second window handles, it is the new window

                if(            (!strWindowHandle.equals(strFirstWindowHandle))       &&                (strWindowHandle.equals(strSecondWindowHandle)    ){

                                //Switch to the new window

                                driver.switchTo().window(strWindowHandle);

                                //Print window title

                                System.out.println(driver.getTitle());

                                break;

                }

}

//Select the item "Third Window" from drop-down list

Select lstSecondWindow = new Select(driver.findElement(By.id("list-second-window")));

lstSecondWindow.selectByVisibleText("Third Window");

//Switch back to first window

driver.switchTo().window(strFirstWindowHandle);

driver.quit();

80. Window handling more than 10 windows:

public void switchToNewWindow(int windowNumber) {

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

Iterator < String > ite = s.iterator();

int i = 1;

while (ite.hasNext() && i < 10) {

String popupHandle = ite.next().toString();

driver.switchTo().window(popupHandle);

System.out.println("Window title is : "+driver.getTitle());

if (i == windowCount) break;

i++;

}

}

81. Read Excel:

String path = "D:\\Book3.xlsx";

//Create File

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

//Create workbook

Workbook wb = WorkbookFactory.*create*(fis);

//Read from sheet1

Sheet s = wb.getSheet("Sheet1");

//Total no of rows

**int** lastRow = s.getLastRowNum();

System.*out*.println("Last row- "+lastRow);

//Declare and row will check upto last

**for**(**int** i=1; i<lastRow; i++){

// Get the rows which present in the excel

Row row = s.getRow(i);

//Count of column

**int** lastCell = row.getLastCellNum();

System.*out*.println(lastCell);

//Declare and column will check upto last

**for**(**int** j=0; j<lastCell; j++){

//Read row and column

Cell RowCol = row.getCell(j);

String value = RowCol.getStringCellValue();

System.*out*.println(value);

}

System.*out*.println();

}

**82. Write Excel:**

String path = "D:\\Book3.xlsx";

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

Workbook wb = WorkbookFactory.*create*(fis);

Sheet s = wb.getSheet("Sheet1");

**int** lastRow = s.getLastRowNum();

**for**(**int** i=1; i<=lastRow; i++){

Row row = s.getRow(i);

Cell cell = row.createCell(2);

cell.setCellValue("Writing Done");

}

FileOutputStream fos = **new** FileOutputStream(path);

wb.write(fos);

fos.close();

}

}

**83. Why are looking for a job change**

**According to me, changes are necessary for everyone to enhance your skills, knowledge and personal growth and financial growth. Your organization is the good platform for me where I can learn more.**

**84. What are the strengths in you**  
**For Strengths we can use as below,  
Adaptability, Hardworking, Honest, Flexibility, Optimistic(positive attitude), Persistence(Regular at work), Self motivated , Fast   Decision making, Never give-up any activity**

**85. Xpath to find particular Text**

driver.findElement(By.*xpath*("//a[contains(.,'Secure Login')]"));-anchor tag with xpath

**86. OOPS concept in selenium:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Definition** | **OOPS** | **Selenium** |
| 1 | Locators in a page class,we utilize these locators in test but we cant see these locators in test. | Abstraction | id,name,xpath etc |
| 2 | Webdriver-Interface Reference variable-driver class-FirefoxDriver | Interface | Webdriver driver=new FirefoxDriver(); |
| 3 | we create base class ,extends in other class | Inheritance | Webdriver waits Property files Excels |
| 4 | Class having multiple methods with same name but different parameters | Method overloading | seconds,MINUTES,Hours |
| 5 | Declaring a method in child class which is already present n parent class | Method over Riding | Get Navigate methods |
| 6 | Binding code and data together in a single unit | Encapsulation | @FindBy WebElement WebDriver FindBy FindElement |

**87. Image comparison:**

**ShutterBug:**

Beauty for shutterbug is Ultimate. It can do many cool features like Taking screenshot for particular element, Comparing different screens in run time, Marking particular element during screenshot , Blurring the image.. etc

<!-- https://mvnrepository.com/artifact/com.assertthat/selenium-shutterbug -->

<dependency>

<groupId>com.assertthat</groupId>

<artifactId>selenium-shutterbug</artifactId>

<version>${version}</version>

</dependency>

**Code:**

public class ShutterbugCompareScreenshot {

@Test

public void capturefullPageScreenshotwithDesiredName()

{

WebDriverManager.chromedriver().setup();

WebDriver driver = new ChromeDriver();

driver.get("http://automationtesting.in");

File image = new File(".\\screenshots\\Expected.png");

BufferedImage expectedImage = ImageIO.read(image);

boolean status = Shutterbug.shootPage(driver,ScrollStrategy.BOTH\_DIRECTIONS,500,true).withName("Actual").equals(expectedImage,0.1);

Assert.assertTrue(status);

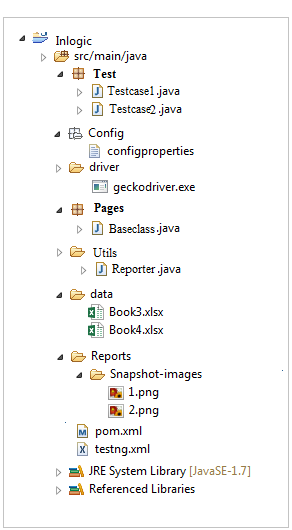
driver.quit();

}

}

**88. About Frame work folder structure:**

**Language:** In our Selenium Project we are using Java language. Even though Selenium supports multiple languages, we are using Java language is just because most of the automation developers have knowledge on Selenium with Java.



**Type of Framework**: Data Driven with TestNG

**Maven:** Using Maven for build, execution and dependency purpose. Integrating the TestNG dependency in POM.xml file and running this POM.xml file using Jenkins

**Version Control Tool:** We use Git as a repository to store our test scripts.

**Jenkins:** By using Jenkins CI (Continuous Integration) Tool, we execute test cases on daily basis and also for nightly execution based on the schedule. Test Result will be sent to the peers using Jenkins.

For example, Home Page and Login Page have a separate classes to store element locators. For the login test there would be a separate class which calls the methods from the Home Page class and Login Page class.

|  |  |  |
| --- | --- | --- |
| **S. No** | **Folder** | **Contains** |
| 1 | Config | Stores the information throughout the framework such as Hub, port and URL. |
| 2 | Data | We use Apache POI to handle excel sheets. we pass test data and handle data driven testing |
| 3 | Driver | browser supports |
| 4 | Utils>Reporter | For the reporting purpose, we are using Extent Reports. It generates beautiful HTML reports. We use the extent reports for maintaining logs and also to include the screenshots of failed test cases in the Extent Report. |
| 5 | Reports>Screenshots-images | Screen shots of a failed test cases will be added in the extent reports. |
| 6 | Base class(Test Folder) | Contains all the common functions initializing web driver, waits and loading config files |
| 7 | Page Folder | All the web page related classes come under **Pages** package  have a separate classes to store element locators |
| 8 | Pom.xml | POM (Project Object Model) is an XML file that contains information about the project and configuration details used by Maven to build the project i.e. source code location, project dependencies etc. |
| 9 | Testng.xml | Using TestNG for Grouping and Parallel execution |
| 10 | Lib | Library files to add the jar files |

**89. Extend Report:**

import org.testng.Assert;

import org.testng.ITestResult;

import org.testng.SkipException;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

import com.aventstack.extentreports.ExtentReports;

import com.aventstack.extentreports.ExtentTest;

import com.aventstack.extentreports.reporter.ExtentHtmlReporter;

public class SampleExtenetReport {

Set the path for our report to generate

**ExtentReports report;**

**ExtentHtmlReporter htmlReporter;**

**ExtentTest logger ===== generate the log**

@BeforeTest

public void startTest() {

report = new ExtentReports();

// Provide the Report Path

htmlReporter = new ExtentHtmlReporter(System.getProperty("user.dir") + "/test-output/myExtentReport.html");

report.attachReporter(htmlReporter);

report.setSystemInfo("Host Name", "Anshul-702HK");

report.setSystemInfo("Env", "Automation Testing");

report.setSystemInfo("User", "Anshul Chauhan");

}

@Test

public void demoTestPass(){

logger = report.createTest("demoTestPass");

Assert.assertTrue(true);

logger.pass("Assert pass as Condition is Pass");

}

@Test

public void demoTestFail(){

logger = report.createTest("demoTestFail");

Assert.assertTrue(false);

logger.pass("Assert Fail as Condition is False");

}

@Test

public void demoTestSkip(){

logger = report.createTest("demoTestSkip");

throw new SkipException("Skipping As Test is not ready");

}

@AfterMethod

public void getTestResult(ITestResult result)**===describe result of a test**

{

if(result.getStatus()==ITestResult.FAILURE)

{

logger.fail(result.getName());

logger.fail(result.getThrowable());

}else if(result.getStatus() == ITestResult.SKIP){

logger.skip("Test Case Skipped is "+result.getName());

}

}

@AfterTest

public void endreport()

{

report.flush();

}

}

**Using TestNG report:**

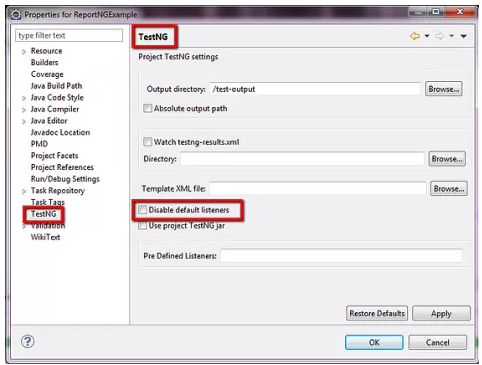
* + Download ReportNG JAR FILES

****

* + Download one more jar files, google juice core jar file



* + Add the jar files in build path
  + To generate the Report NG report we need to disable TestNG default listeners
  + Right click the project>Select “Properties”
  + In the list, click on the option “TestNG”
  + Check the check box “Disable default listeners”
  + Click on Apply and click on Ok button



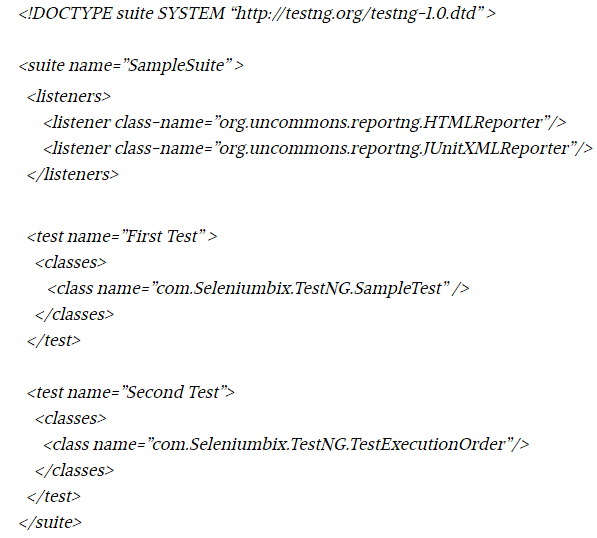
Need to add 2 listeners in the test.xml

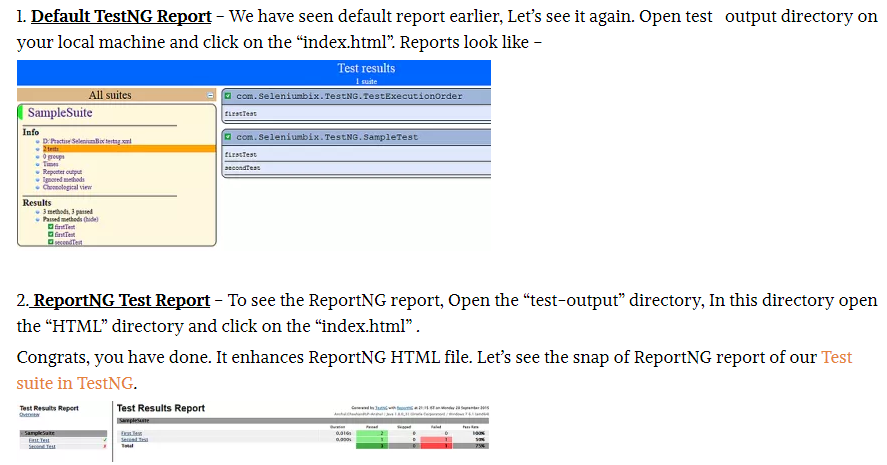
<listeners>

<listener class-name=”org.uncommoms.reportng.HTMLReporter”/>

<listener class-name=”org.uncommoms.reportng.JUnitXMLReporter”/>

</listeners>





**90. Get total no of rows in a table:**

WebElement table = driver.findElement(By.xpath("//table[@class='gf-table']"));

List<WebElement> row = table.findElements(By.tagName("tr"));

System.out.println("Total Number of Rows = " + row.size());

**Print content of rows:**

for (int i=0;i<=row.size()-1 ;i++){

System.out.println(row.get(i).getText());

}

**91. Get total no of columns in a table:**

List<WebElement> column = row.get(10).findElements(By.tagName("td"));

System.out.println("Total Number of Column = " + column.size());

**Print content of 2nd Column**

List<WebElement> firstcol = driver.findElements(By.xpath("//table[@class='gf-table']/tbody/tr/td[2]"));

for (int j=0; j<=row.size();j++){

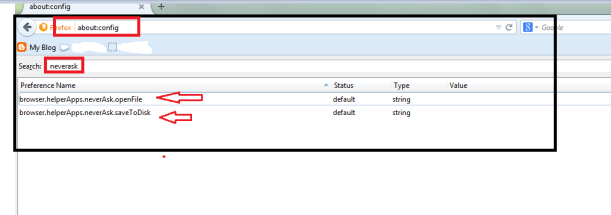
System.out.println(firstcol.get(j).getText());

}

**92. Download a file:**

## How to Download files in Selenium Webdriver

1- Open [Firefox](http://learn-automation.com/selenium-script-firefox/) browser and in url box type about:config and press enter

2- In Search bar type neverask and enter, here you will find some settings (refer below screenshot)  
[](https://i1.wp.com/learn-automation.com/wp-content/uploads/2015/03/116.png)

Now you can see here value is blank so we need to mention which type of file it will not ask if download starts in case.

Note- In this post I am giving values for .exe file(application), in your case if you want to download pdf, excel file etc. you need to mention values (MIME type).

Here you can find their MIME type that will be the values for these settings.

<http://www.sitepoint.com/web-foundations/mime-types-complete-list/>

## How to Download files in Selenium Webdriver

Step 1- Create a firefox Profile.

Step 2- set Preferences as per requirement.

Step 3- Open Firefox with firefox profile.

Let us implement the same through Script.

Note- This script will download [Adobe](http://www.adobe.com/) Reader from [Filehippo.com](http://filehippo.com/)

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.firefox.FirefoxProfile;

public class DownloadFiles {

public static void main(String[] args) {

// Create a profile

FirefoxProfile profile=new FirefoxProfile();

// Set preferences for file type

profile.setPreference("browser.helperApps.neverAsk.openFile", "application/octet-stream");

// Open browser with profile

WebDriver driver=new FirefoxDriver(profile);

// Set implicit wait

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

// Maximize window

driver.manage().window().maximize();

// Open APP to download application

driver.get("http://www.filehippo.com/download\_adobe\_reader");

// Click on download

driver.findElement(By.xpath(".//\*[@id='program-header']/div[4]/a[1]")).click();

}

}

**Compare 2 Excel sheet data:**

FileInputstream f1=new FileInputstream(“path”);

FileInputstream f2=new FileInputstream(“path”);

Workbook wb1=WorkbookFactory.create(f1);

Workbook wb2=WorkbookFactory.create(f2);

Sheet s1=wb1.getSheet(“Sheetname1”);

Sheet s2=wb2.getSheet(“Sheetname2”);

For(int i=2;i<7;i++)

{

Row row1=s1.getRow(i);

Cell Cell1=row1.getcell(6);

s.o.pln(cell1);

}

For(int j=5;j<10;j++)

{

Row row2=s2.getRow(j);

Cell Cell2=row2.getcell(26);

s.o.pln(cell2);

}

If(cell1==cell2)

{

s.o.pln(“true”)

}

Else{

s.o.pln(“false”)

}

Comapre Data in check box value match in the table:

Driver.findElement(By.id(“checkbox”)).click();

Driver.findElement(By.cssselctor(“”)).click();

String data=driver.findELement(By.Id(“”)).gettext();

s.o.pln(“Result is” +data);

string expecteddata=”Bike”;

if(data.equals(expecteddata))

{

s.o.pln(“Matched”)

{

Else

{

s.o.pln(“unMatched”);

}

}

Maximize the browser using Desired Capabilities:

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

**Using a Chrome executable in a non-standard location**

ChromeOptions options = new ChromeOptions();

options.setBinary("/path/to/other/chrome/binary");