**Note: practise with code**

1.    what are limitations in selenium webdriver

Ans :

·         It doesn’t support non- web based applications, it only supports web based applications.

·         No inbuilt reporting capability reporting capability so we need to use plugins like Junit and TestNG for test reports.

·         Selenium partially supports for Dialog boxes.

·         Selenium does not support file upload facility.

2.    installing/configure selenium

Ans :

·         Install java in our computer.

·         Install Eclipse in our computer

·         Launch eclipse then create new workspace.

·         Go to selenium official website and download Java client library.

·         Navigate towards Eclipse IDE. Create a new java project based project following File-> New-> Java Projects.

·         Create a new Java class under the source folder by right clicking on it and navigating to New -> class.

·         Select the project and Right click on it. Select “Properties” within the listed options. Select “Java Build Path”

·         By default, “Libraries” tab is opened. If not, click on the “Libraries” tab. Then, click on the “Add External Jars…” button. Browse to the location where we have saved the extracted folder for Java Client Libraries

·         Select all the JAR files present in the “selenium-java” folder and click on open button within the dialog box

·         Click on the “OK” button within the dialog box so as to complete the configuration part of Selenium Libraries in our java project.

3.    what are different ways of locating elements in selenium

        Ans:

Browser tools to identify web elements are

·         Firebug for firefox, FirePath and selenium IDE

·         Google developer tools for chrome

·         Web inspector for safari

Different possibilities to identify elements by:

·         By ID

driver.findElement(By.id("element id"))

·         By Class

driver.findElement(By.className("element class"))

·         By Name

driver.findElement(By.name("element name"))

·         By CSS selector

driver.findElement(By.cssSelector("css selector"))

·         By Xpath

driver.findElement(By.xpath("xpath expression"))

·         By Tag name

driver.findElement(By.tagName("element html tag name"))

·         By Link :

driver.findElement(By.link("link text"))

Which is fastest way to identify elements in web page?

Ans: ID selectors are best and fastest way. IDs are the safest, fastest locator option and should always be your first choice. IDs should be unique in every page.

What is absolute path and relative path in xpath?

**Absolute Xpath**: It uses complete path from the Root Element to the desire element.

WebElement link=driver.findElement(By.xpath("/html/body/li[@id='test']/a"));

**Relative Xpath**: You can simply start by referencing the element you want and go from there.

WebElement link = driver.findElement(By.xpath(".//\*[@id='test']/a"));

Different types of waits or synchronization in selenium webdriver

Ans : Two types of waits :

**Explicit waits (Unconditional):**

Ex: wait() and Thread.sleep();

We can tell the tool to wait only till the Condition met. Once the condition is met, the tool proceed with the next step.

WebDriverWait wait = new WebDriverWait(driver, 10);

wait.until(ExpectedConditions.visibilityOf(username));

Thread.sleep waits the specified time irrespective of the object state.

Thread.sleep(2000);

**Implicit waits (Conditional):**

An implicit wait is to tell WebDriver to poll the DOM for a certain amount of time when trying to find an element or elements if they are not immediately available.

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

how to save screen shots using selenium webdriver

Ans:

File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

FileUtils.copyFile(scrFile, new File("C:\\screenshot.png"));

how to handle multiple windows in selenium webdriver

Ans :

// Opening site

    driver.findElement(By.xpath("//img[@alt='SeleniumMasterLogo']")).click();

    // Storing parent window reference into a String Variable

    String Parent\_Window = driver.getWindowHandle();

     // Switching from parent window to child window

**for** (String Child\_Window : driver.getWindowHandles())

    {

    driver.switchTo().window(Child\_Window);

    // Performing actions on child window

}

// Switching back to Parent Window

    driver.switchTo().window(Parent\_Window);

how to lanuch webpage using chrome driver

Ans :

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

          //open the browser

          WebDriver dr= **new** ChromeDriver();

          Thread.sleep(3000);

          // open facebook page https://www.facebook.com/

          dr.get("https://www.facebook.com/");

what is desired capabilities in selenium webdriver

Ans :

The Desired Capabilities Class helps us to tell the webdriver, which environment we are going to use in our test script. The testing environment can be a web browser, mobile device, mobile emulator, mobile simulator, etc.

It gives facility to set the properties of browser. Such as to set BrowserName, Platform, Version of Browser.

Ex:

DesiredCapabilities capability = DesiredCapabilities.firefox();

capability.setBrowserName("firefox");

capability.setPlatform(Platform.WIN8\_1);

driver = new RemoteWebDriver(new URL(nodeUrl),capability);

how to set language while opening website

Ex:

Firefox :

FirefoxProfile profile = new FirefoxProfile();

//setting the locale french : ‘fr’

profile.setPreference(“intl.accept\_languages”,”fr”);

driver = new FirefoxDriver(profile);

Chrome :

System.setProperty(“webdriver.chrome.driver”,”D:/DollarArchive/chromedriver.exe”);

ChromeOptions options = new ChromeOptions();

options.addArguments(“–lang= sl”);

ChromeDriver driver = new ChromeDriver(options);

**how to handle windows based popups (upload and dropdown)**

Ex: write code

write code to verify any application login page is working or not

(u should write code to use textbox, button click events)

Ex:

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

          //open the browser

          WebDriver dr= **new** ChromeDriver();

          Thread.sleep(3000);

          // open facebook page https://www.facebook.com/

          dr.get("https://www.facebook.com/");

          Thread.sleep(3000);

          // enter the user name

          dr.findElement(By.id("email")).sendKeys("neelima.sindhu@gmail.com");

          Thread.sleep(3000);

          // enter the password

          dr.findElement(By.id("pass")).sendKeys("xxxxxx");

          Thread.sleep(3000);

          //click on login button

          dr.findElement(By.xpath(".//\*[@id='loginbutton']")).click();

how to select items from dropdown/select box

Ex:

//open the browser

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

          Thread.sleep(3000);

          // open the mercury tours website

          d.get("http://www.newtours.demoaut.com/");

          Thread.sleep(3000);

          // enter the username

          d.findElement(By.name("userName")).sendKeys("mahesh");

          Thread.sleep(3000);

          // enter the password

          d.findElement(By.name("password")).sendKeys("mahesh");

          Thread.sleep(3000);

      // click on sign in image button

      d.findElement(By.xpath("html/body/div[1]/table/tbody/tr/td[2]/table/tbody/tr[4]/td/table/tbody/tr/td[2]/table/tbody/tr[2]/td[3]/form/table/tbody/tr[4]/td/table/tbody/tr[4]/td[2]/div/input")).click();

          Thread.sleep(3000);

          // select the no of passengers

          Select sel= **new** Select( d.findElement(By.name("passCount")));

          Thread.sleep(3000);

          // select the values as 2 passengers

          sel.selectByIndex(1);

          Thread.sleep(3000);

          // Select the value as 1 passenger

          sel.selectByValue("1");

          Thread.sleep(3000);

          // select the destination place

          Select selDest= **new** Select( d.findElement(By.name("toPort")));

          Thread.sleep(3000);

          // select the destination as London

          selDest.selectByValue("London");

          Thread.sleep(3000);

          // select the destination as  Frankfurt

          selDest.selectByIndex(1);

          Thread.sleep(3000);

          // close the browser

          d.close();

how to know if checkbox is checked or not in webpage

ex:

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

          dr.get("http://www.keynotesupport.com/websites/contact-form-example-checkboxes.shtml");

          WebElement we=dr.findElement(By.*xpath*(".//\*[@id='align']/span[5]/input"));

**if**(!(we.isSelected())){

                 System.***out***.println("The check box is not checked");

                 we.click();

          }

tell me code to pass values from parent window to child window

Ex: write code

write code to find out if all links are working or not

Ex:

**public** **void** validateInvalidLinks() {

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

                 driver.get("http://google.com");

**try** {

                       invalidLinksCount = 0;

                       List<WebElement> anchorTagsList = driver.findElements(By

                                     .tagName("a"));

                       System.out.println("Total no. of links are "

                                     + anchorTagsList.size());

**for** (WebElement anchorTagElement : anchorTagsList) {

**if** (anchorTagElement != **null**) {

                                     String url = anchorTagElement.getAttribute("href");

**if** (url != **null** && !url.contains("javascript")) {

                                            verifyURLStatus(url);

                                     } **else** {

                                            invalidLinksCount++;

                                     }

                              }

                       }

                       System.out.println("Total no. of invalid links are "

                                     + invalidLinksCount);

                 } **catch** (Exception e) {

                       e.printStackTrace();

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

                 }

          }

**public** **void** verifyURLStatus(String URL) {

                 HttpClient client = HttpClientBuilder.create().build();

                 HttpGet request = **new** HttpGet(URL);

**try** {

                       HttpResponse response = client.execute(request);

                       // verifying response code and The HttpStatus should be 200 if not,

                       // increment invalid link count

                       ////We can also check for 404 status code like response.getStatusLine().getStatusCode() == 404

**if** (response.getStatusLine().getStatusCode() != 200)

                              invalidLinksCount++;

                 } **catch** (Exception e) {

                       e.printStackTrace();

                 }

          }

Write code on how to use javascriptexecutor?

Ex:

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript(Script,Arguments);

difference between assert and verify?

Ans : When an “assert” fails, the test will be aborted.Where if a “verify” fails, the test will continue executing and logging the failure.

Ex: // verification

**public** **void** verification\_method() {

           String time = driver.findElement(By.*xpath*("//div[@id='timeLeft']")).getText();

**if** (time == "Tuesday, 28 January 2014")

           {

            System.***out***.print("\nText Match");

           }

**else**

           {

            System.***out***.print("\nText does Match");

           }

          }

   }

  // assertion :

**public** **void** assertion\_method() {

           Actualtext = driver.findElement(By.xpath("//h2/span")).getText();

           Assert.assertEquals(Actualtext, "Tuesday, 28 January 2014");

           System.out.print("\n assertion\_method\_1() -> Part executed");

          }

difference between driver.close and driver.quit methods?

driver.close – It closes the browser window on which the focus is set.

driver.quit – It basically calls driver.dispose method which in turn closes all the browser windows and ends the WebDriver session gracefully.

Ex:  driver.close();

   Driver.quit();

common exceptions in selenium?

1.    ElementNotVisibleException:

Thrown when an element is present on the DOM, but it is not visible, and so is not able to be interacted with.Most commonly encountered when trying to click or read text of an element that is hidden from view.

2.    ElementNotSelectableException:

Thrown when trying to select an unselectable element.

    3. NoSuchElement : An element could not be located on the page using the given search parameters

how to handle Ajax calls in selenium?

Ex:

WebDriverWait wait = new WebDriverWait(driver, waitTime); wait.until(ExpectedConditions.visibilityOfElementLocated(locator));

**public** **void** test\_AjaxExample() {

                 /\*Wait for grid to appear\*/

                 By container = By.cssSelector(".demo-container");

                 wait = **new** WebDriverWait(driver, 5);

                 wait.until(ExpectedConditions.presenceOfElementLocated(container));

                 /\*Get the text before performing an ajax call\*/

                 WebElement noDatesTextElement = driver.findElement(By.xpath("//div[@class='RadAjaxPanel']/span"));

                 String textBeforeAjaxCall = noDatesTextElement.getText().trim();

                 /\*Click on the date\*/

                 driver.findElement(By.linkText("1")).click();

                 /\*Wait for loader to disappear \*/

                 By loader = By.className("raDiv");

                 wait.until(ExpectedConditions.invisibilityOfElementLocated(loader));

                 /\*Get the text after ajax call\*/

                 WebElement selectedDatesTextElement = driver.findElement(By.xpath("//div[@class='RadAjaxPanel']/span"));

                 wait.until(ExpectedConditions.visibilityOf(selectedDatesTextElement));

                 String textAfterAjaxCall = selectedDatesTextElement.getText().trim();

                 /\*Verify both texts before ajax call and after ajax call text.\*/

                 Assert.assertNotEquals(textBeforeAjaxCall, textAfterAjaxCall);

                 String expectedTextAfterAjaxCall = "Thursday, January 01, 2015";

                 /\*Verify expected text with text updated after ajax call\*/

                 Assert.assertEquals(textAfterAjaxCall, expectedTextAfterAjaxCall);

          }

we have webtable, need to click on second row from table.

when we click on child will be populated. first column in primary column for each row.

tell me steps to verify child form has proper data or not

Ex:

How to assign the value to textbox other than sendkeys method?

Ans :

|  |  |
| --- | --- |
|  | Try to set the element's value using the executeScript method:  driver.executeScript(“document.getElementById(‘elementID’).setAttribute(‘value’,’newvalue for the element’)”); |

Selenium grid, how to execute scripts on multiple browser

s