## WebDriver Code

public class PG1 {

public static void main(String[] args) {

// declaration and instantiation of objects/variables

System.setProperty("webdriver.gecko.driver","C:\\geckodriver.exe");

WebDriver driver = new FirefoxDriver();

//comment the above 2 lines and uncomment below 2 lines to use Chrome

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

//WebDriver driver = new ChromeDriver();

String baseUrl = "http://demo.guru99.com/test/newtours/";

String expectedTitle = "Welcome: Mercury Tours";

String actualTitle = "";

// launch Fire fox and direct it to the Base URL

driver.get(baseUrl);

// get the actual value of the title

actualTitle = driver.getTitle();

if (actualTitle.contentEquals(expectedTitle)){

System.out.println("Test Passed!");

} else {

System.out.println("Test Failed");

}

//close Fire fox

driver.close();

}

}

## Locating GUI Elements

package newproject;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class PG2 {

public static void main(String[] args) {

System.setProperty("webdriver.gecko.driver","C:\\geckodriver.exe");

WebDriver driver = new FirefoxDriver();

String baseUrl = "http://www.facebook.com";

String tagName = "";

driver.get(baseUrl);

tagName = driver.findElement(By.id("email")).getTagName();

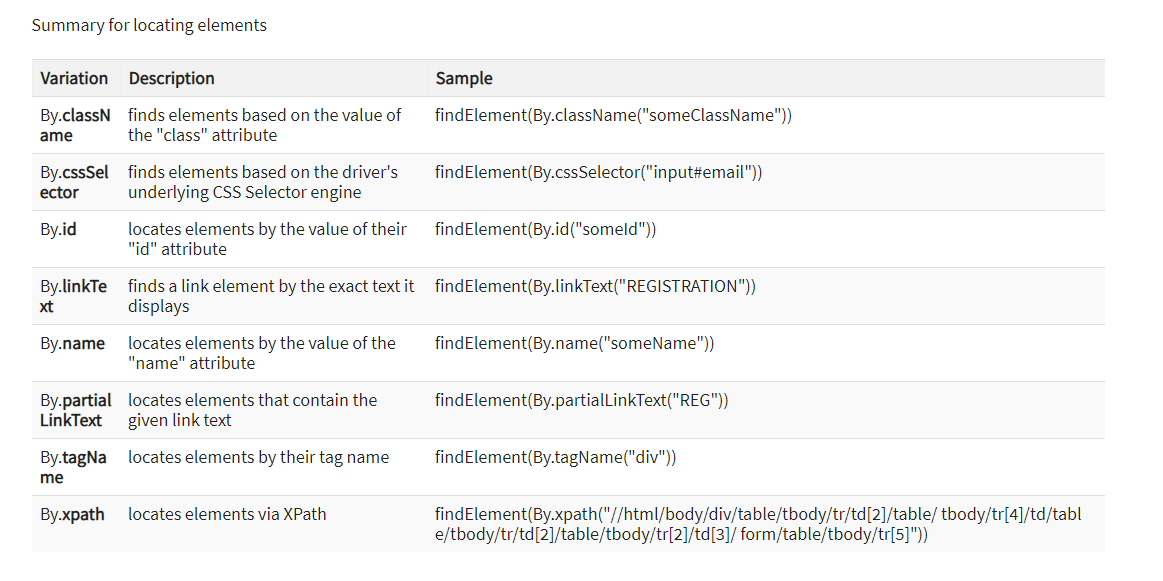
System.out.println(tagName);

driver.close();

System.exit(0);

}

}



**Switching Between Frames**

package newproject;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class PG4 {

public static void main(String[] args) {

System.setProperty("webdriver.gecko.driver","C:\\geckodriver.exe");

WebDriver driver = new FirefoxDriver();

driver.get("http://demo.guru99.com/selenium/deprecated.html");

driver.switchTo().frame("classFrame");

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

driver.close();

}

}

## Switching Between Pop-up Windows

package mypackage;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class myclass {

public static void main(String[] args) {

System.setProperty("webdriver.gecko.driver","C:\\geckodriver.exe");

WebDriver driver = new FirefoxDriver();

String alertMessage = "";

driver.get("http://jsbin.com/usidix/1");

driver.findElement(By.cssSelector("input[value=\"Go!\"]")).click();

alertMessage = driver.switchTo().alert().getText();

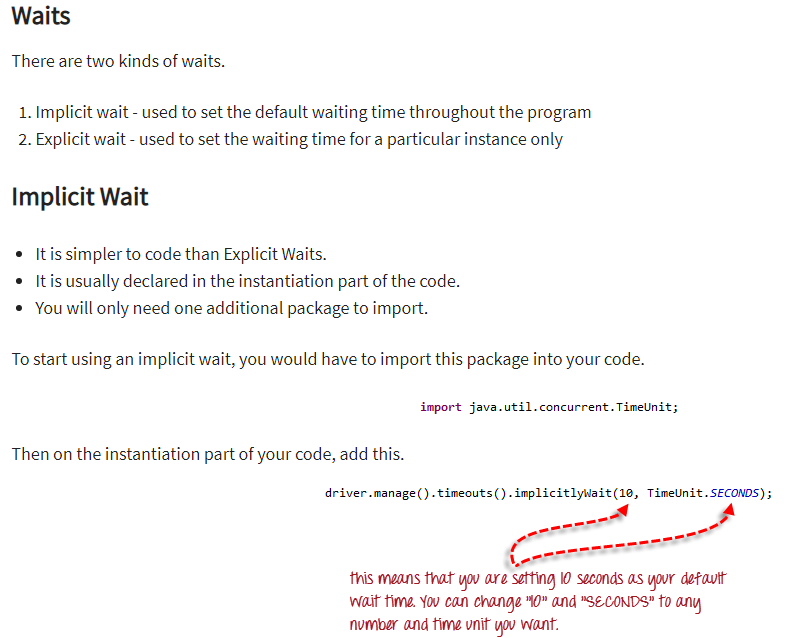
driver.switchTo().alert().accept();

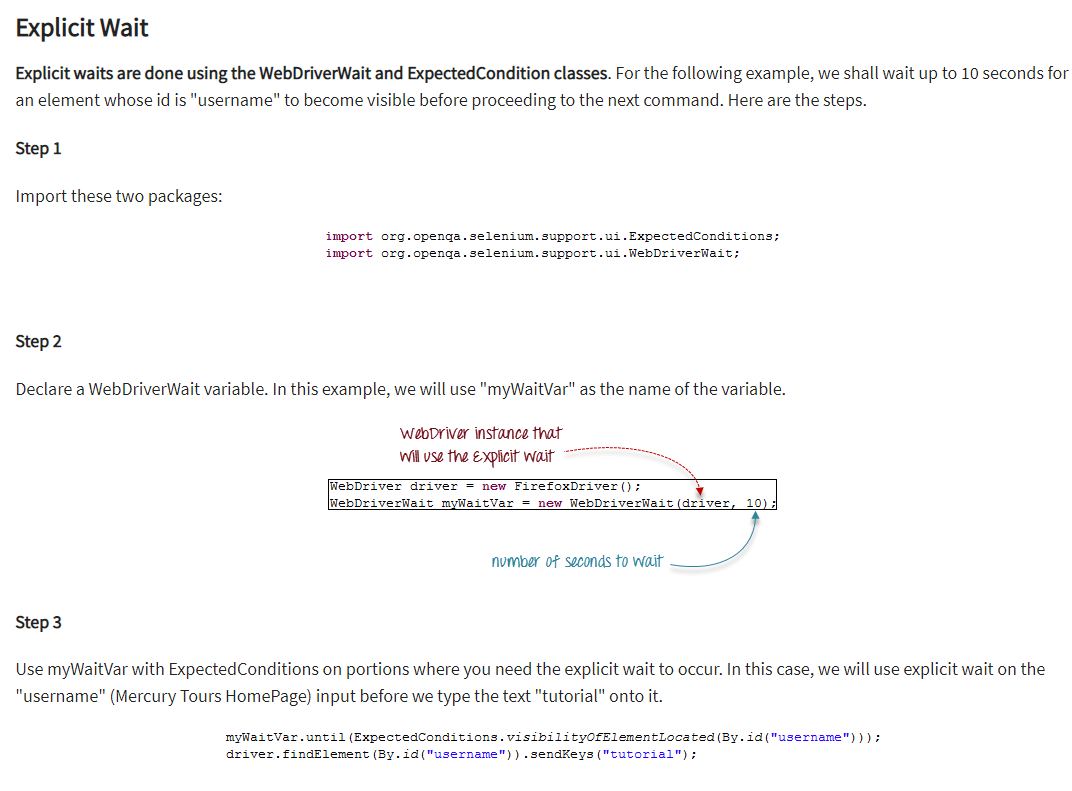
System.out.println(alertMessage);

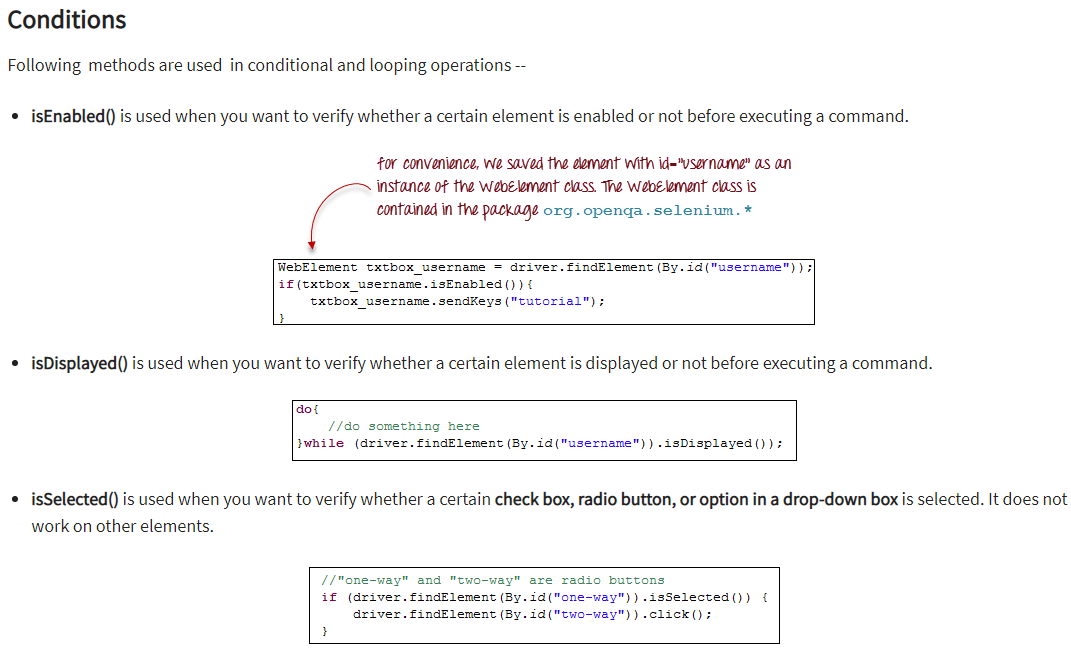
driver.quit();

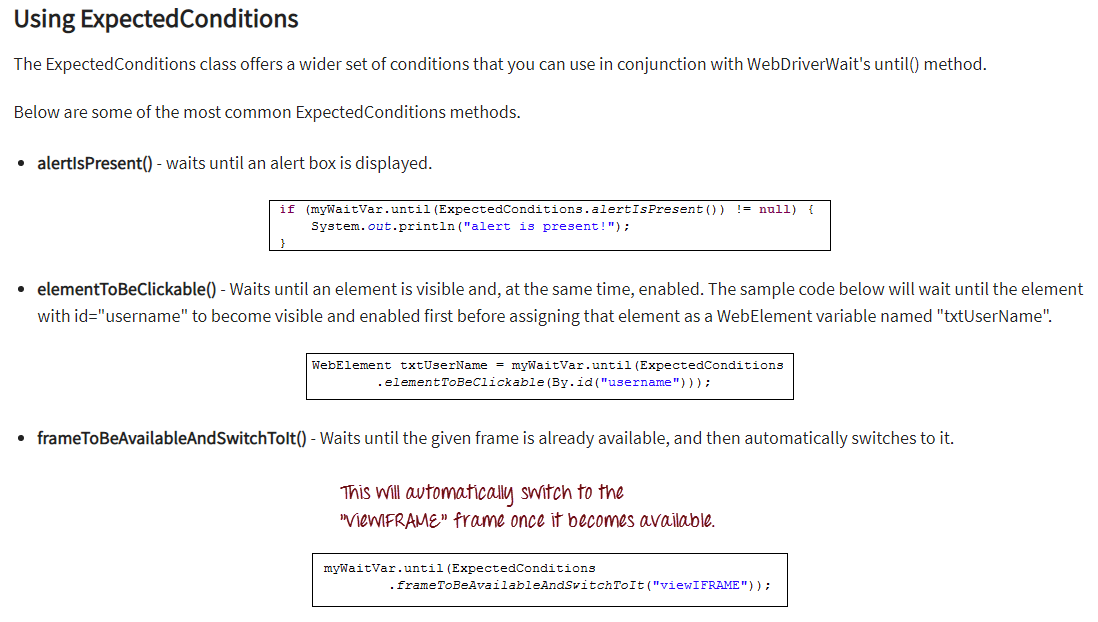
}

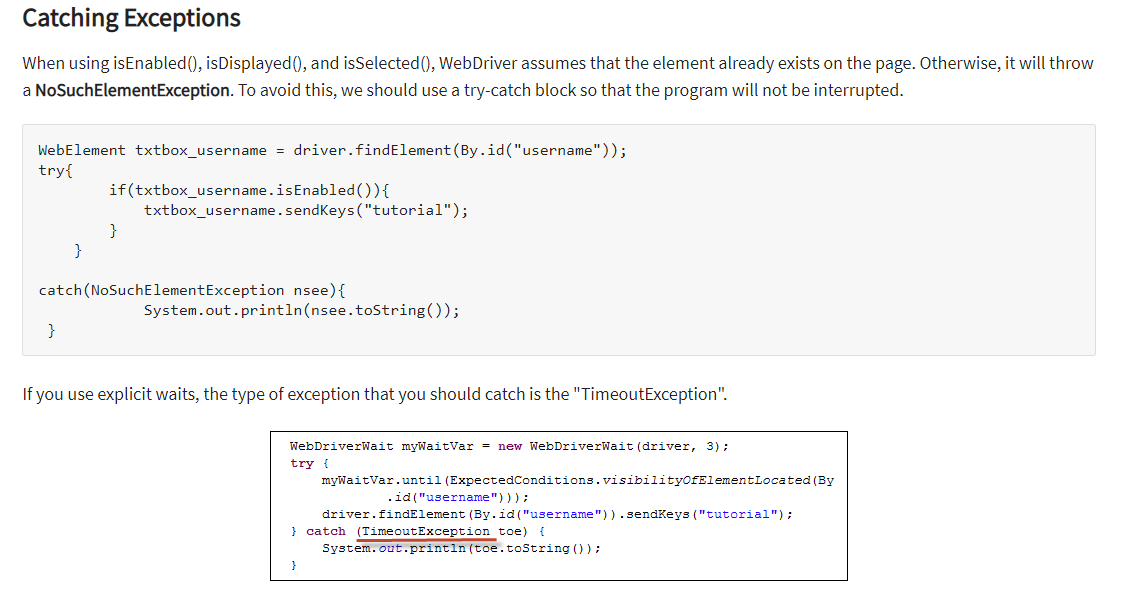
}











## FindElements

WebElement loginLink = driver.findElement(By.linkText("Login"));

List<WebElement> elementName = driver.findElements(By.LocatorStrategy("LocatorValue"));

List<WebElement> listOfElements = driver.findElements(By.xpath("//div"));

package com.sample.stepdefinitions;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class NameDemo {

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver", "D:\\3rdparty\\chrome\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

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

driver.get("http://demo.guru99.com/test/ajax.html");

// Find the radio button for “No” using its ID and click on it

driver.findElement(By.id("no")).click();

//Click on Check Button

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

//List elements

List<WebElement> elements = driver.findElements(By.name("name"));

System.out.println("Number of elements:" +elements.size());

for (int i=0; i<elements.size();i++){

System.out.println("Radio button text:" + elements.get(i).getAttribute("value"));

}

}

}

# Selenium Form WebElement: TextBox, Submit Button, sendkeys(), click()

public class Form {

public static void main(String[] args) {

// declaration and instantiation of objects/variables

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

WebDriver driver = new ChromeDriver();

String baseUrl = "http://demo.guru99.com/test/login.html";

driver.get(baseUrl);

// Get the WebElement corresponding to the Email Address(TextField)

WebElement email = driver.findElement(By.id("email"));

// Get the WebElement corresponding to the Password Field

WebElement password = driver.findElement(By.name("passwd"));

email.sendKeys("[abcd@gmail.com](mailto:abcd@gmail.com)");

password.sendKeys("abcdefghlkjl");

System.out.println("Text Field Set");

// Deleting values in the text box

email.clear();

password.clear();

System.out.println("Text Field Cleared");

// Find the submit button

WebElement login = driver.findElement(By.id("SubmitLogin"));

// Using click method to submit form

email.sendKeys("[abcd@gmail.com](mailto:abcd@gmail.com)");

password.sendKeys("abcdefghlkjl");

login.click();

System.out.println("Login Done with Click");

//using submit method to submit the form. Submit used on password field

driver.get(baseUrl);

driver.findElement(By.id("email")).sendKeys("[abcd@gmail.com](mailto:abcd@gmail.com)");

driver.findElement(By.name("passwd")).sendKeys("abcdefghlkjl");

driver.findElement(By.id("SubmitLogin")).submit();

System.out.println("Login Done with Submit");

}

}

# How to Select CheckBox and Radio Button in Selenium WebDriver

public class Form {

public static void main(String[] args) {

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

WebDriver driver = new ChromeDriver();

driver.get("http://demo.guru99.com/test/radio.html");

WebElement radio1 = driver.findElement(By.id("vfb-7-1"));

WebElement radio2 = driver.findElement(By.id("vfb-7-2"));

//Radio Button1 is selected

radio1.click();

System.out.println("Radio Button Option 1 Selected");

//Radio Button1 is de-selected and Radio Button2 is selected

radio2.click();

System.out.println("Radio Button Option 2 Selected");

// Selecting CheckBox

WebElement option1 = driver.findElement(By.id("vfb-6-0"));

// This will Toggle the Check box

option1.click();

// Check whether the Check box is toggled on

if (option1.isSelected()) {

System.out.println("Checkbox is Toggled On");

} else {

System.out.println("Checkbox is Toggled Off");

}

//Selecting Checkbox and using isSelected Method

driver.get("http://demo.guru99.com/test/facebook.html");

WebElement chkFBPersist = driver.findElement(By.id("persist\_box"));

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

chkFBPersist.click ();

System.out.println("Facebook Persists Checkbox Status is - "+chkFBPersist.isSelected());

}

}

}

# How to Click on Image in Selenium Webdriver

public class MyClass {

public static void main(String[] args) {

String baseUrl = "https://www.facebook.com/login/identify?ctx=recover";

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

WebDriver driver = new ChromeDriver();

driver.get(baseUrl);

//click on the "Facebook" logo on the upper left portion

driver.findElement(By.cssSelector("a[title=\"Go to Facebook home\"]")).click();

//verify that we are now back on Facebook's homepage

if (driver.getTitle().equals("Facebook - log in or sign up")) {

System.out.println("We are back at Facebook's homepage");

} else {

System.out.println("We are NOT in Facebook's homepage");

}

}

}

# How to Select Value from DropDown using Selenium Webdriver

public class accessDropDown {

public static void main(String[] args) {

System.setProperty("webdriver.gecko.driver","C:\\geckodriver.exe");

String baseURL = "http://demo.guru99.com/test/newtours/register.php";

WebDriver driver = new FirefoxDriver();

driver.get(baseURL);

Select drpCountry = new Select(driver.findElement(By.name("country")));

drpCountry.selectByVisibleText("ANTARCTICA");

//Selecting Items in a Multiple SELECT elements

driver.get("http://jsbin.com/osebed/2");

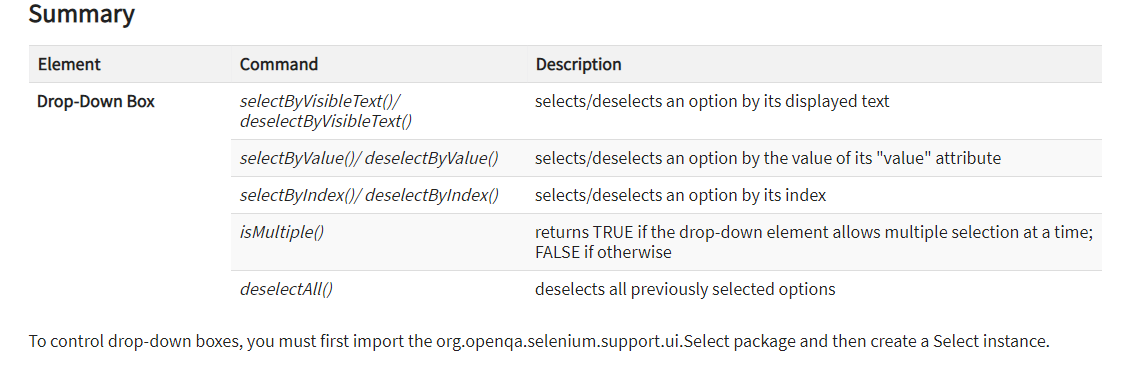
Select fruits = new Select(driver.findElement(By.id("fruits")));

fruits.selectByVisibleText("Banana");

fruits.selectByIndex(1);

}

}



# Locate Elements by Link Text & Partial Link Text in Selenium Webdriver

public class MyClass {

public static void main(String[] args) {

String baseUrl = "http://demo.guru99.com/test/link.html";

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

WebDriver driver = new ChromeDriver();

driver.get(baseUrl);

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

System.out.println("title of page is: " + driver.getTitle()); driver.findElement(By.partialLinkText("here")).click();

System.out.println("Title of page is: " + driver.getTitle());

}

}

# Mouse Click & Keyboard Event: Action Class in Selenium Webdriver

|  |  |
| --- | --- |
| **Method** | **Description** |
| **clickAndHold()** | Clicks (without releasing) at the current mouse location. |
| **contextClick()** | Performs a context-click at the current mouse location. (Right Click Mouse Action) |
| **doubleClick()** | Performs a double-click at the current mouse location. |
| **dragAndDrop(source, target)** | Performs click-and-hold at the location of the source element, moves to the location of the target element, then releases the mouse. |
| **dragAndDropBy(source, x-offset, y-offset)** | Performs click-and-hold at the location of the source element, moves by a given offset, then releases the mouse. |
| **keyDown(modifier\_key)** | Performs a modifier key press. Does not release the modifier key - subsequent interactions may assume it's kept pressed |
| **keyUp(modifier \_key)** | Performs a key release. |
| **moveByOffset(x-offset, y-offset)** | Moves the mouse from its current position (or 0,0) by the given offset.. |
| **moveToElement(toElement)** | Moves the mouse to the middle of the element. |
| **release()** | Releases the depressed left mouse button at the current mouse location |
| **sendKeys(onElement, charsequence)** | Sends a series of keystrokes onto the element. |

import org.openqa.selenium.interactions.Action;

import org.openqa.selenium.interactions.Actions;

public class PG7 {

public static void main(String[] args) {

String baseUrl = "http://demo.guru99.com/test/newtours/";

System.setProperty("webdriver.gecko.driver","C:\\geckodriver.exe");

WebDriver driver = new FirefoxDriver();

driver.get(baseUrl);

WebElement link\_Home = driver.findElement(By.linkText("Home"));

WebElement td\_Home = driver

.findElement(By

.xpath("//html/body/div"

+ "/table/tbody/tr/td"

+ "/table/tbody/tr/td"

+ "/table/tbody/tr/td"

+ "/table/tbody/tr"));

Actions builder = new Actions(driver);

Action mouseOverHome = builder

.moveToElement(link\_Home)

.build();

String bgColor = td\_Home.getCssValue("background-color");

System.out.println("Before hover: " + bgColor);

mouseOverHome.perform();

bgColor = td\_Home.getCssValue("background-color");

System.out.println("After hover: " + bgColor);

driver.close();

}

}

public static void main(String[] args) {

String baseUrl = "http://www.facebook.com/";

WebDriver driver = new FirefoxDriver();

driver.get(baseUrl);

WebElement txtUsername = driver.findElement(By.id("email"));

Actions builder = new Actions(driver);

Action seriesOfActions = builder

.moveToElement(txtUsername)

.click()

.keyDown(txtUsername, Keys.SHIFT)

.sendKeys(txtUsername, "hello")

.keyUp(txtUsername, Keys.SHIFT)

.doubleClick(txtUsername)

.contextClick()

.build();

seriesOfActions.perform() ;

}

# How to Upload & Download a File using Selenium Webdriver

public class PG8 {

public static void main(String[] args) {

System.setProperty("webdriver.gecko.driver","C:\\geckodriver.exe");

String baseUrl = "http://demo.guru99.com/test/yahoo.html";

WebDriver driver = new FirefoxDriver();

driver.get(baseUrl);

WebElement downloadButton = driver.findElement(By

.id("messenger-download"));

String sourceLocation = downloadButton.getAttribute("href");

String wget\_command = "cmd /c C:\\Wget\\wget.exe -P D: --no-check-certificate " + sourceLocation;

try {

Process exec = Runtime.getRuntime().exec(wget\_command);

int exitVal = exec.waitFor();

System.out.println("Exit value: " + exitVal);

} catch (InterruptedException | IOException ex) {

System.out.println(ex.toString());

}

driver.close();

}

}

# XPath in Selenium WebDriver Tutorial: How to Find XPath?

Xpath=//input[@type='text']

Xpath= //label[@id='message23']

Xpath= //input[@value='RESET']

Xpath=//\*[@class='barone']

Xpath=//a[@href='http://demo.guru99.com/']

Xpath= //img[@src='//cdn.guru99.com/images/home/java.png']

Xpath=//\*[contains(@type,'sub')]

Xpath=//\*[contains(@name,'btn')]

Xpath=//\*[contains(text(),'here')]

Xpath=//\*[@type='submit' or @name='btnReset']

Xpath=//input[@type='submit' and @name='btnLogin']

Xpath=//label[starts-with(@id,'message')]

Xpath=//\*[@type='text']//following::input[1]

Xpath=//\*[text()='Enterprise Testing']//ancestor::div

Xpath=//\*[@id='java\_technologies']//child::li

Xpath=//\*[@type='submit']//preceding::input

xpath=//\*[@type='submit']//following-sibling::input

Xpath=//\*[@id='rt-feature']//parent::div

Xpath =//\*[@type='password']//self::input

Xpath=//\*[@id='rt-feature']//descendant::a

# Alert & Popup Window Handling in Selenium WebDriver

1) void dismiss() **// To click on the 'Cancel' button of the alert.**

driver.switchTo().alert().dismiss();

2) void accept() **// To click on the 'OK' button of the alert.**

driver.switchTo().alert().accept();

3) String getText**() // To capture the alert message.**

driver.switchTo().alert().getText();

4) void sendKeys(String stringToSend) **// To send some data to alert box.**

driver.switchTo().alert().sendKeys("Text");

import org.openqa.selenium.NoAlertPresentException;

import org.openqa.selenium.Alert;

public class AlertDemo {

public static void main(String[] args) throws NoAlertPresentException,InterruptedException {

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

WebDriver driver = new ChromeDriver();

// Alert Message handling

driver.get("http://demo.guru99.com/test/delete\_customer.php");

driver.findElement(By.name("cusid")).sendKeys("53920");

driver.findElement(By.name("submit")).submit();

// Switching to Alert

Alert alert = driver.switchTo().alert();

// Capturing alert message.

String alertMessage= driver.switchTo().alert().getText();

// Displaying alert message

System.out.println(alertMessage);

Thread.sleep(5000);

// Accepting alert

alert.accept();

}

}

**Handling multiple windows in selenium webdriver using above scenario.**

public class WindowHandle\_Demo {

public static void main(String[] args) throws InterruptedException {

WebDriver driver=new FirefoxDriver();

//Launching the site.

driver.get("http://demo.guru99.com/popup.php");

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

driver.findElement(By.xpath("//\*[contains(@href,'popup.php')]")).click();

String MainWindow=driver.getWindowHandle();

// To handle all new opened window.

Set<String> s1=driver.getWindowHandles();

Iterator<String> i1=s1.iterator();

while(i1.hasNext())

{

String ChildWindow=i1.next();

if(!MainWindow.equalsIgnoreCase(ChildWindow))

{

// Switching to Child window

driver.switchTo().window(ChildWindow);

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

.sendKeys("[gaurav.3n@gmail.com](mailto:gaurav.3n@gmail.com)

driver.findElement(By.name("btnLogin")).click();

// Closing the Child Window.

driver.close();

}

}

// Switching to Parent window i.e Main Window.

driver.switchTo().window(MainWindow);

}

}

## Using X-Path to Locate Web Table Elements

## 

## Get all the values of a Dynamic Table



# How to Verify Tooltip using Selenium WebDriver

public class ToolTip {

public static void main(String[] args) {

String baseUrl = "http://demo.guru99.com/test/social-icon.html";

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

WebDriver driver = new ChromeDriver();

driver.get(baseUrl);

String expectedTooltip = "Github";

// Find the Github icon at the top right of the header

WebElement github = driver.findElement(By.xpath(".//\*[@class='soc-ico show-round']/a[4]"));

//get the value of the "title" attribute of the github icon

String actualTooltip = github.getAttribute("title");

//Assert the tooltip's value is as expected

System.out.println("Actual Title of Tool Tip"+actualTooltip);

if(actualTooltip.equals(expectedTooltip)) {

System.out.println("Test Case Passed");

}

String baseUrl = "http://demo.guru99.com/test/tooltip.html";

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

WebDriver driver = new ChromeDriver();

String expectedTooltip = "What's new in 3.2";

driver.get(baseUrl);

WebElement download = driver.findElement(By.xpath(".//\*[@id='download\_now']"));

Actions builder = new Actions (driver);

builder.clickAndHold().moveToElement(download);

builder.moveToElement(download).build().perform();

WebElement toolTipElement = driver.findElement(By.xpath(".//\*[@class='box']/div/a"));

String actualTooltip = toolTipElement.getText();

System.out.println("Actual Title of Tool Tip "+actualTooltip);

if(actualTooltip.equals(expectedTooltip)) {

System.out.println("Test Case Passed"); }

driver.close();

}

}

package com.group.guru99;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.Assert;

import org.testng.annotations.Test;

public class TC\_Class1 {

public static final WebDriver webDriver = new FirefoxDriver();;

String launchPageHeading = "//h3[text()='Guru99 Bank']";

final String userName\_element = "//input[@name='uid']", password\_element = "//input[@name='password']",

signIn\_element = "//input[@name='btnLogin']";

final String userName\_value = "mngr28642", password\_value = "ydAnate";

final String managerID = "//td[contains(text(),'Manger Id')]";

final String newCustomer = "//a[@href='addcustomerpage.php']", fundTransfer = "//a[@href='FundTransInput.php']";

/\*\*

\* This test case will initialize the webDriver

\*/

@Test(groups = { "bonding", "strong\_ties" })

public void tc01LaunchURL() {

webDriver.manage().window().maximize();

webDriver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

webDriver.get("http://www.demo.guru99.com/V4/");

}

/\*\*

\* Will check the presence of Heading on Login Page

\*/

@Test(groups = { "bonding" })

public void tc02VerifyLaunchPage() {

Assert.assertTrue(webDriver.findElement(By.xpath(launchPageHeading)).isDisplayed(),

"Home Page heading is not displayed");

System.out.println("Home Page heading is displayed");

}

/\*\*

\* This test case will enter User name, password and will then click on

\* signIn button

\*/

@Test(groups = { "bonding", "strong\_ties" })

public void tc03EnterCredentials() {

webDriver.findElement(By.xpath(userName\_element)).sendKeys(userName\_value);

webDriver.findElement(By.xpath(password\_element)).sendKeys(password\_value);

webDriver.findElement(By.xpath(signIn\_element)).click();

}

/\*\*

\* This test case will verify manger's ID presence on DashBoard

\*/

@Test(groups = { "strong\_ties" })

public void tc04VerifyLoggedInPage() {

Assert.assertTrue(webDriver.findElement(By.xpath(managerID)).isDisplayed(),

"Manager ID label is not displayed");

System.out.println("Manger Id label is displayed");

}

/\*\*

\* This test case will check the presence of presence of New customer link

\* And FundTransfer link in Left pannel

\*/

@Test(groups = { "bonding" })

public void tc05VerifyHyperlinks() {

Assert.assertTrue(webDriver.findElement(By.xpath(newCustomer)).isEnabled(),

"New customer hyperlink is not displayed");

System.out.println("New customer hyperlink is displayed");

Assert.assertTrue(webDriver.findElement(By.xpath(fundTransfer)).isEnabled(),

"Fund Transfer hyperlink is not displayed");

System.out.println("Fund Transfer hyperlink is displayed");

}

}

## What is Page Object Model?

**Page Object Model (POM)** is a design pattern, popularly used in test automation that creates Object Repository for web UI elements. The advantage of the model is that it reduces code duplication and improves test maintenance.

Under this model, for each web page in the application, there should be a corresponding Page Class. This Page class will identify the WebElements of that web page and also contains Page methods which perform operations on those WebElements. Name of these methods should be given as per the task they are performing, i.e., if a loader is waiting for the payment gateway to appear, POM method name can be waitForPaymentScreenDisplay().

**Advantages of POM**

1. Page Object Design Pattern says operations and flows in the UI should be separated from verification. This concept makes our code cleaner and easy to understand.
2. The Second benefit is the object repository is independent of test cases, so we can use the same object repository for a different purpose with different tools. For example, we can integrate Page Object Model in Selenium with TestNG/JUnit for functional[Testing](https://www.guru99.com/software-testing.html)and at the same time with JBehave/Cucumber for acceptance testing.
3. Code becomes less and optimized because of the reusable page methods in the POM classes.
4. Methods get more realistic names which can be easily mapped with the operation happening in UI. i.e. if after clicking on the button we land on the home page, the method name will be like 'gotoHomePage()'.

Here are we are dealing with 2 pages

1. Login Page
2. Home Page (shown once you login)

Accordingly, we create 2 POM in Selenium classes

**Guru99 Login page POM**

package pages;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

public class Guru99Login {

WebDriver driver;

By user99GuruName = By.name("uid");

By password99Guru = By.name("password");

By titleText =By.className("barone");

By login = By.name("btnLogin");

public Guru99Login(WebDriver driver){

this.driver = driver;

}

//Set user name in textbox

public void setUserName(String strUserName){

driver.findElement(user99GuruName).sendKeys(strUserName);

}

//Set password in password textbox

public void setPassword(String strPassword){

driver.findElement(password99Guru).sendKeys(strPassword);

}

//Click on login button

public void clickLogin(){

driver.findElement(login).click();

}

//Get the title of Login Page

public String getLoginTitle(){

return driver.findElement(titleText).getText();

}

public void loginToGuru99(String strUserName,String strPasword){

//Fill user name

this.setUserName(strUserName);

//Fill password

this.setPassword(strPasword);

//Click Login button

this.clickLogin();

}

}

**Guru99 Home Page POM in Selenium**

package pages;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

public class Guru99HomePage {

WebDriver driver;

By homePageUserName = By.xpath("//table//tr[@class='heading3']");

public Guru99HomePage(WebDriver driver){

this.driver = driver;

}

//Get the User name from Home Page

public String getHomePageDashboardUserName(){

return driver.findElement(homePageUserName).getText();

}

}

**Guru99 Simple POM in Selenium Test case**

package test;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.Assert;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

import pages.Guru99HomePage;

import pages.Guru99Login;

public class Test99GuruLogin {

String driverPath = "C:\\geckodriver.exe";

WebDriver driver;

Guru99Login objLogin;

Guru99HomePage objHomePage;

@BeforeTest

public void setup(){

System.setProperty("webdriver.gecko.driver", driverPath);

driver = new FirefoxDriver();

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

driver.get("http://demo.guru99.com/V4/");

}

@Test(priority=0)

public void test\_Home\_Page\_Appear\_Correct(){

//Create Login Page object

objLogin = new Guru99Login(driver);

//Verify login page title

String loginPageTitle = objLogin.getLoginTitle();

Assert.assertTrue(loginPageTitle.toLowerCase().contains("guru99 bank"));

//login to application

objLogin.loginToGuru99("mgr123", "mgr!23");

// go the next page

objHomePage = new Guru99HomePage(driver);

//Verify home page

Assert.assertTrue(objHomePage.getHomePageDashboardUserName().toLowerCase().contains("manger id : mgr123"));

}

# How to Select Date from DatePicker/Calendar in Selenium Webdriver

public class DatePicker {

@Test

public void testDAtePicker() throws Exception{

//DAte and Time to be set in textbox

String dateTime ="12/07/2014 2:00 PM";

WebDriver driver = new FirefoxDriver();

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

driver.get("https://demos.telerik.com/kendo-ui/datetimepicker/index");

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

//button to open calendar

WebElement selectDate = driver.findElement(By.xpath("//span[@aria-controls='datetimepicker\_dateview']"));

selectDate.click();

//button to move next in calendar

WebElement nextLink = driver.findElement(By.xpath("//div[@id='datetimepicker\_dateview']//div[@class='k-header']//a[contains(@class,'k-nav-next')]"));

//button to click in center of calendar header

WebElement midLink = driver.findElement(By.xpath("//div[@id='datetimepicker\_dateview']//div[@class='k-header']//a[contains(@class,'k-nav-fast')]"));

//button to move previous month in calendar

WebElement previousLink = driver.findElement(By.xpath("//div[@id='datetimepicker\_dateview']//div[@class='k-header']//a[contains(@class,'k-nav-prev')]"));

//Split the date time to get only the date part

String date\_dd\_MM\_yyyy[] = (dateTime.split(" ")[0]).split("/");

//get the year difference between current year and year to set in calander

int yearDiff = Integer.parseInt(date\_dd\_MM\_yyyy[2])- Calendar.getInstance().get(Calendar.YEAR);

midLink.click();

if(yearDiff!=0){

//if you have to move next year

if(yearDiff>0){

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

System.out.println("Year Diff->"+i);

nextLink.click();

}

}

//if you have to move previous year

else if(yearDiff<0){

for(int i=0;i< (yearDiff\*(-1));i++){

System.out.println("Year Diff->"+i);

previousLink.click();

}

}

}

Thread.sleep(1000);

//Get all months from calendar to select correct one

List<WebElement> list\_AllMonthToBook = driver.findElements(By.xpath("//div[@id='datetimepicker\_dateview']//table//tbody//td[not(contains(@class,'k-other-month'))]"));

list\_AllMonthToBook.get(Integer.parseInt(date\_dd\_MM\_yyyy[1])-1).click();

Thread.sleep(1000);

//get all dates from calendar to select correct one

List<WebElement> list\_AllDateToBook = driver.findElements(By.xpath("//div[@id='datetimepicker\_dateview']//table//tbody//td[not(contains(@class,'k-other-month'))]"));

list\_AllDateToBook.get(Integer.parseInt(date\_dd\_MM\_yyyy[0])-1).click();

///FOR TIME

WebElement selectTime = driver.findElement(By.xpath("//span[@aria-controls='datetimepicker\_timeview']"));

//click time picker button

selectTime.click();

//get list of times

List<WebElement> allTime = driver.findElements(By.xpath("//div[@data-role='popup'][contains(@style,'display: block')]//ul//li[@role='option']"));

dateTime = dateTime.split(" ")[1]+" "+dateTime.split(" ")[2];

//select correct time

for (WebElement webElement : allTime) {

if(webElement.getText().equalsIgnoreCase(dateTime))

{

webElement.click();

}

}

}

}

# Data Driven, Keyword Driven & Hybrid

### ReadGuru99ExcelFile.java

package excelExportAndFileIO;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;

import org.apache.poi.ss.usermodel.Sheet;

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

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

public class ReadGuru99ExcelFile {

public Sheet readExcel(String filePath,String fileName,String sheetName) throws IOException{

//Create a object of File class to open xlsx file

File file = new File(filePath+"\\"+fileName);

//Create an object of FileInputStream class to read excel file

FileInputStream inputStream = new FileInputStream(file);

Workbook guru99Workbook = null;

//Find the file extension by spliting file name in substing and getting only extension name

String fileExtensionName = fileName.substring(fileName.indexOf("."));

//Check condition if the file is xlsx file

if(fileExtensionName.equals(".xlsx")){

//If it is xlsx file then create object of XSSFWorkbook class

guru99Workbook = new XSSFWorkbook(inputStream);

}

//Check condition if the file is xls file

else if(fileExtensionName.equals(".xls")){

//If it is xls file then create object of XSSFWorkbook class

guru99Workbook = new HSSFWorkbook(inputStream);

}

//Read sheet inside the workbook by its name

Sheet guru99Sheet = guru99Workbook.getSheet(sheetName);

return guru99Sheet;

}

}

### ReadObject.java

package operation;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import java.io.InputStream;

import java.util.Properties;

public class ReadObject {

Properties p = new Properties();

public Properties getObjectRepository() throws IOException{

//Read object repository file

InputStream stream = new FileInputStream(new File(System.getProperty("user.dir")+"\\src\\objects\\object.properties"));

//load all objects

p.load(stream);

return p;

}

}

### UIOperation.java

public class UIOperation {

WebDriver driver;

public UIOperation(WebDriver driver){

this.driver = driver;

}

public void perform(Properties p,String operation,String objectName,String objectType,String value) throws Exception{

System.out.println("");

switch (operation.toUpperCase()) {

case "CLICK":

//Perform click

driver.findElement(this.getObject(p,objectName,objectType)).click();

break;

case "SETTEXT":

//Set text on control

driver.findElement(this.getObject(p,objectName,objectType)).sendKeys(value);

break;

case "GOTOURL":

//Get url of application

driver.get(p.getProperty(value));

break;

case "GETTEXT":

//Get text of an element

driver.findElement(this.getObject(p,objectName,objectType)).getText();

break;

default:

break;

}

}

/\*\*

\* Find element BY using object type and value

\* @param p

\* @param objectName

\* @param objectType

\* @return

\* @throws Exception

\*/

private By getObject(Properties p,String objectName,String objectType) throws Exception{

//Find by xpath

if(objectType.equalsIgnoreCase("XPATH")){

return By.xpath(p.getProperty(objectName));

}

//find by class

else if(objectType.equalsIgnoreCase("CLASSNAME")){

return By.className(p.getProperty(objectName));

}

//find by name

else if(objectType.equalsIgnoreCase("NAME")){

return By.name(p.getProperty(objectName));

}

//Find by css

else if(objectType.equalsIgnoreCase("CSS")){

return By.cssSelector(p.getProperty(objectName));

}

//find by link

else if(objectType.equalsIgnoreCase("LINK")){

return By.linkText(p.getProperty(objectName));

}

//find by partial link

else if(objectType.equalsIgnoreCase("PARTIALLINK")){

return By.partialLinkText(p.getProperty(objectName));

}else

{

throw new Exception("Wrong object type");

}

}

}

### ExecuteTest.java

public class ExecuteTest {

@Test

public void testLogin() throws Exception {

// TODO Auto-generated method stub

WebDriver webdriver = new FirefoxDriver();

ReadGuru99ExcelFile file = new ReadGuru99ExcelFile();

ReadObject object = new ReadObject();

Properties allObjects = object.getObjectRepository();

UIOperation operation = new UIOperation(webdriver);

//Read keyword sheet

Sheet guru99Sheet = file.readExcel(System.getProperty("user.dir")+"\\","TestCase.xlsx" , "KeywordFramework");

//Find number of rows in excel file

int rowCount = guru99Sheet.getLastRowNum()-guru99Sheet.getFirstRowNum();

//Create a loop over all the rows of excel file to read it

for (int i = 1; i < rowCount+1; i++) {

//Loop over all the rows

Row row = guru99Sheet.getRow(i);

//Check if the first cell contain a value, if yes, That means it is the new testcase name

if(row.getCell(0).toString().length()==0){

//Print testcase detail on console

System.out.println(row.getCell(1).toString()+"----"+ row.getCell(2).toString()+"----"+

row.getCell(3).toString()+"----"+ row.getCell(4).toString());

//Call perform function to perform operation on UI

operation.perform(allObjects, row.getCell(1).toString(), row.getCell(2).toString(),

row.getCell(3).toString(), row.getCell(4).toString());

}

else{

//Print the new testcase name when it started

System.out.println("New Testcase->"+row.getCell(0).toString() +" Started");

}

}

}

}

Read database

public class SQLConnector {

public static void main(String[] args) throws ClassNotFoundException, SQLException {

//Connection URL Syntax: "jdbc:mysql://ipaddress:portnumber/db\_name"

String dbUrl = "jdbc:mysql://localhost:3036/emp";

//Database Username

String username = "root";

//Database Password

String password = "guru99";

//Query to Execute

String query = "select \* from employee;";

//Load mysql jdbc driver

Class.forName("com.mysql.jdbc.Driver");

//Create Connection to DB

Connection con = DriverManager.getConnection(dbUrl,username,password);

//Create Statement Object

Statement stmt = con.createStatement();

// Execute the SQL Query. Store results in ResultSet

ResultSet rs= stmt.executeQuery(query);

// While Loop to iterate through all data and print results while (rs.next()){

String myName = rs.getString(1);

String myAge = rs.getString(2);

System. out.println(myName+" "+myAge);

}

// closing DB Connection

con.close();

}

}

**Guru99CrossBrowserScript.java**

public class CrossBrowserScript {

WebDriver driver;

/\*\*

\* This function will execute before each Test tag in testng.xml

\* @param browser

\* @throws Exception

\*/

@BeforeTest

@Parameters("browser")

public void setup(String browser) throws Exception{

//Check if parameter passed from TestNG is 'firefox'

if(browser.equalsIgnoreCase("firefox")){

//create firefox instance

System.setProperty("webdriver.gecko.driver", ".\\geckodriver.exe");

driver = new FirefoxDriver();

}

//Check if parameter passed as 'chrome'

else if(browser.equalsIgnoreCase("chrome")){

//set path to chromedriver.exe

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

//create chrome instance

driver = new ChromeDriver();

}

//Check if parameter passed as 'Edge'

else if(browser.equalsIgnoreCase("Edge")){

//set path to Edge.exe

System.setProperty("webdriver.edge.driver",".\\MicrosoftWebDriver.exe");

//create Edge instance

driver = new EdgeDriver();

}

else{

//If no browser passed throw exception

throw new Exception("Browser is not correct");

}

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

}

@Test

public void testParameterWithXML() throws InterruptedException{

driver.get("http://demo.guru99.com/V4/");

//Find user name

WebElement userName = driver.findElement(By.name("uid"));

//Fill user name

userName.sendKeys("guru99");

//Find password

WebElement password = driver.findElement(By.name("password"));

//Fill password

password.sendKeys("guru99");

}

}

# How to Take Screenshot in Selenium WebDriver

public class Guru99TakeScreenshot {

@Test

public void testGuru99TakeScreenShot() throws Exception{

WebDriver driver ;

System.setProperty("webdriver.gecko.driver","C:\\geckodriver.exe");

driver = new FirefoxDriver();

driver.get("http://demo.guru99.com/V4/");

//Call take screenshot function

this.takeSnapShot(driver, "c://test.png") ;

}

/\*\*

\* This function will take screenshot

\* @param webdriver

\* @param fileWithPath

\* @throws Exception

\*/

public static void takeSnapShot(WebDriver webdriver,String fileWithPath) throws Exception{

//Convert web driver object to TakeScreenshot

TakesScreenshot scrShot =((TakesScreenshot)webdriver);

//Call getScreenshotAs method to create image file

File SrcFile=scrShot.getScreenshotAs(OutputType.FILE);

//Move image file to new destination

File DestFile=new File(fileWithPath);

//Copy file at destination

FileUtils.copyFile(SrcFile, DestFile);

}

}

public class TestScreenshotUsingAshot {

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

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

WebDriver driver = new ChromeDriver();

driver.get("http://demo.guru99.com/test/guru99home/");

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

Screenshot = new AShot().shootingStrategy(ShootingStrategies.viewportPasting(1000)).takeScreenshot(driver);

ImageIO.write(screenshot.getImage(), "jpg", new File("c:\\ElementScreenshot.jpg"));

}

}

## Taking a screenshot of a particular element of the page

public class TestElementScreenshotUsingAshot {

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

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

WebDriver driver = new ChromeDriver();

driver.get("http://demo.guru99.com/test/guru99home/");

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

// Find the element to take a screenshot

WebElement element = driver.findElement(By.xpath ("//\*[@id=\"site-name\"]/a[1]/img"));

// Along with driver pass element also in takeScreenshot() method.

Screenshot = new AShot().shootingStrategy(ShootingStrategies.viewportPasting(1000)).takeScreenshot(driver,element);

ImageIO.write(screenshot.getImage(), "jpg", new File("c:\\ElementScreenshot.jpg"));

}

}

## Image Comparison using AShot

public class TestImageComaprison {

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

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

WebDriver driver = new ChromeDriver();

driver.get("http://demo.guru99.com/test/guru99home/");

// Find the element and take a screenshot

WebElement logoElement = driver.findElement(By.xpath("//\*[@id=\"site-name\"]/a[1]/img"));

Screenshot logoElementScreenshot = new AShot().takeScreenshot(driver, logoElemnent);

// read the image to compare

BufferedImage expectedImage = ImageIO.read(new File("C:\\Guru99logo.png"));

BufferedImage actualImage = logoElementScreenshot.getImage();

// Create ImageDiffer object and call method makeDiff()

ImageDiffer imgDiff = new ImageDiffer();

ImageDiff diff = imgDiff.makeDiff(actualImage, expectedImage);

if (diff.hasDiff() == true) {

System.out.println("Images are same");

} else {

System.out.println("Images are different");

}

driver.quit();

}

}