Q1- 1 - Open http://seleniumpractise.blogspot.com/2016/08/how-to-use-explicit-wait-in-selenium.html

Click on timer

Wait for text "WebDriver"

ANS:

package base;

import java.time.Duration;

import java.util.List;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

import org.openqa.selenium.edge.EdgeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

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

import org.openqa.selenium.support.ui.Select;

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

import org.testng.Assert;

public class Assignment11 {

static WebDriver driver;

static boolean SignUpBtn;

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

// TODO Auto-generated method stub

ChromeOptions opt=new ChromeOptions();

opt.addArguments("--remote-allow-origins=\*");

//Launching the browser

driver=new ChromeDriver(opt);

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

//Pass the URL

driver.get("http://seleniumpractise.blogspot.com/2016/08/how-to-use-explicit-wait-in-selenium.html");

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

//Click me Start Timer

driver.findElement(By.xpath("//button[contains(text(),'Click me to start timer')]")).click();

//wait for webDriver element

WebDriverWait wait=new WebDriverWait(driver, Duration.ofSeconds(20));

wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//p[text()='WebDriver']")));

String WebDriverElement=driver.findElement(By.xpath("//p[text()='WebDriver']")).getText();

System.out.println("The Expected Element Found and Element name is "+" " +WebDriverElement);

Assert.assertEquals(WebDriverElement, "WebDriver","WebDriver Element Found");

driver.close();

}

}

Q2-

2- Create a method in Utility class named as "captureScreenshotOfWebElement"

will capture the screenshot of element

will store element screenshot in element screenshot folder

all screenshot should have date time stamp

ANS:

Utility Class::::

package base;

import java.io.File;

import java.io.IOException;

import java.util.Date;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.By;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.ui.ExpectedCondition;

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

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

public class Utility

{

static WebDriver driver;

public static void Takescreenshotofelement(By Locator) throws IOException

{

//For Date Stamp in screenshot

Date d = new Date();

String Datestamp = d.toString().replace(":", "\_").replace(" ", "\_") + ".png";

//ScreenShot Method

// Converting the web driver object to TakeScreenshot

TakesScreenshot scrensht=((TakesScreenshot)driver);

File sourceFile=scrensht.getScreenshotAs(OutputType.FILE);

//Storing Image to specific location

File SpecificLocation=new File("/Practice/src/test/resources/Screenshots" +Datestamp);

//Copy to new location

FileUtils.copyFile(sourceFile, SpecificLocation);

}

}

Test Class:::

package base;

import java.io.File;

import java.io.IOException;

import java.time.Duration;

import java.util.List;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

import org.openqa.selenium.edge.EdgeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

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

import org.openqa.selenium.support.ui.Select;

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

import org.testng.Assert;

public class Assignment11 {

static WebDriver driver;

static boolean SignUpBtn;

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

// TODO Auto-generated method stub

ChromeOptions opt=new ChromeOptions();

opt.addArguments("--remote-allow-origins=\*");

//Launching the browser

driver=new ChromeDriver(opt);

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

//Pass the URL

driver.get("http://seleniumpractise.blogspot.com/2016/08/how-to-use-explicit-wait-in-selenium.html");

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

//Click me Start Timer

By web=By.xpath("//button[contains(text(),'Click me to start timer')]");

//Taking a screenshot of the element

Utility.Takescreenshotofelement(web);

}

}

Q3- Create a method in Utility class named as "waitForElement" (dont use any waits)

method should except locator and timeout in seconds

will wait until element is not enabled

add sleep of 1 second

handle all exception while retrying

ANS:

Utility Class::::

package base;

import java.io.File;

import java.io.IOException;

import java.time.Duration;

import java.util.Date;

import java.util.NoSuchElementException;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.By;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.ui.ExpectedCondition;

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

import org.openqa.selenium.support.ui.FluentWait;

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

public class Utility

{

static WebDriver driver;

static org.openqa.selenium.support.ui.FluentWait<WebDriver> wait;

public static void WaitforElementSelect(By Locator,int timeout) throws IOException

{

wait=FluentWait(driver).withTimeout(Duration.ofSeconds(30)).pollingEvery(Duration.ofSeconds(5)).ignoring(NoSuchElementException.class);

wait=new WebDriverWait(driver, Duration.ofSeconds(timeout));

wait.until(ExpectedConditions.visibilityOfElementLocated(Locator));

}

private static FluentWait<WebDriver> FluentWait(WebDriver driver2) {

return null;

}

}

TestClass::::

package base;

import java.io.File;

import java.io.IOException;

import java.time.Duration;

import java.util.List;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

import org.openqa.selenium.edge.EdgeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

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

import org.openqa.selenium.support.ui.Select;

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

import org.testng.Assert;

public class Assignment11 {

static WebDriver driver;

static boolean SignUpBtn;

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

// TODO Auto-generated method stub

ChromeOptions opt=new ChromeOptions();

opt.addArguments("--remote-allow-origins=\*");

//Launching the browser

driver=new ChromeDriver(opt);

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

//Pass the URL

driver.get("http://seleniumpractise.blogspot.com/2016/08/how-to-use-explicit-wait-in-selenium.html");

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

//Click me Start Timer

By web=By.xpath("//button[contains(text(),'Click me to start timer')]");

//Taking screenshot of the element

Utility.WaitforElementSelect(web, 20);

}

}