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

Click on timer

Wait for text "WebDriver"

Use 2 conditions of ExpectedCondition

package assignment11;

import java.time.Duration;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

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

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

public class Task1 {

public static void main(String[] args) {

/\*

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

\* Click on timer

\* Wait for text "WebDriver" Use 2 conditions of ExpectedCondition

\*

\*

\*/

WebDriver driver = new ChromeDriver();

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

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

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

WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(15));

// Method 1 - textToBePresentInElementLocated

//Boolean status = wait.until(ExpectedConditions.textToBePresentInElementLocated(By.xpath("//p[@id='demo']"), "WebDriver"));

// Method 2 - textToBe

Boolean status=wait.until(ExpectedConditions.textToBe(By.xpath("//p[@id='demo']"), "WebDriver"));

if(status==true) {

System.out.println("Pass");

}else {

System.out.println("Fail");

}

driver.quit();

}

}

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

package assignment11;

import java.io.File;

import java.io.IOException;

import java.text.SimpleDateFormat;

import java.time.Duration;

import java.util.Date;

import org.openqa.selenium.By;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.io.FileHandler;

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

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

public class Task2 {

/\*

\* 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

\*

\*/

public static void captureScreenshotOfWebElement(WebElement element) {

File src = element.getScreenshotAs(OutputType.FILE);

File dest = new File("D:\\SeleniumTrainingWorkspace\\SeleniumWebDriver\\screenshots\\ElementScreenshot\_"

+ getCustomDateFormat() + ".png");

try {

FileHandler.copy(src, dest);

} catch (IOException e) {

System.out.println("Something went wrong-" + e.getMessage());

}

}

public static String getCustomDateFormat() {

Date date = new Date();

SimpleDateFormat dateFormat = new SimpleDateFormat("HH\_mm\_ss\_dd\_MM\_yyyy");

String currentDate = dateFormat.format(date);

return currentDate;

}

}

4- 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

package assignment11;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Utility {

public static void main(String[] args) {

// TODO Auto-generated method stub

WebDriver driver = new ChromeDriver();

driver.get("");

waitForElement(driver,By.name(""),10);

}

public static void waitForElement(WebDriver driver,By locator, int timeOut) {

WebElement element = driver.findElement(locator);

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

try {

element.isEnabled();

}catch(Exception e){

System.out.println("Wating for element to enabled"+e.getMessage());

waitForSeconds(1);

}

}

}

public static void waitForSeconds(int seconds) {

try {

Thread.sleep(seconds \* 1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}