**Webdriver Waits**

Synchronizing a Test Using Waits

One of the Important Factor in test automation for a complex web application is to ensure that the flow of the test cases should be in synchronization with the application under test (AUT).

When tests are run, the application may not always respond with the same speed. For example, it might take a few seconds for a progress bar to reach 100 percent, a status message to appear, a button to become enabled, and a window or pop-up message to open. We can handle these anticipated timing problems by synchronizing our test to ensure that Selenium WebDriver waits until our application is ready before performing a certain action. There are several options that we can use to synchronize our test.

**Implicitwait**

Syntax: driver.manage().timeouts().implicitlyWait(long time, TimeUnit.***SECONDS***);

Specifies the amount of time the driver should wait when searching for an element if it is not immediately present.

When searching for a single element, the driver should poll the page until the element has been found, or this timeout expires before throwing a NoSuchElementException. When searching for multiple elements, the driver should poll the page until at least one element has been found or this timeout has expired.

Increasing the implicit wait timeout should be used judiciously as it will have an adverse effect on test run time, especially when used with slower location strategies like XPath.

Parameters:

time: The amount of time to wait.

Unit: The unit of measure for time.

Returns: A self reference.

**Explicit Wait:**

When an Explicit Wait is implemented it provides a better control when compared with an implicit wait. Unlike an implicit wait, we can write custom code or conditions for wait before proceeding further in the code. An explicit wait can only be implemented in cases where synchronization is needed and the rest of the script is working fine. The Selenium WebDriver provides **WebDriverWait** and **ExpectedCondition** classes for implementing an explicit wait. The ExpectedCondition class provides a set of predefined conditions to wait before proceeding further in the code.

The following are some common conditions that we frequently come across when automating web browsers supported by the ExpectedCondition class

|  |  |
| --- | --- |
| **Predefined condition** | **Selenium method** |
| An element is visible and enabled | elementToBeClickable(By locator) |
| An element is selected | elementToBeSelected(WebElement element) |
| Presence of an element | presenceOfElementLocated(By locator) |
| Specific text present in an element | textToBePresentInElement(By locator, java.lang.String text) |

**How it Works:-**

We can create a wait for a set of common conditions using the ExpectedCondition class. First, we need to create an instance of the WebDriverWait class by passing the driver instance and timeout for a wait as follows:

WebDriverWait wait = **new** WebDriverWait(driver, 10);

Next, ExpectedCondition is passed to the wait.until() method as follows:

wait.until(ExpectedConditions.titleContains("selenium"));

Note: - The WebDriverWait object will call the ExpectedCondition class object every 500 milliseconds until it returns successfully.

**// pageLoadTimeout**

// Sets the amount of time to wait for a page load to complete before throwing an error. If the timeout is negative, page loads can be indefinite.

**// setScriptTimeout**

// Sets the amount of time to wait for an asynchronous script to finish execution before throwing an error. If the timeout is negative, then the script will be allowed to run indefinitely.

**Example**:

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.Alert;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.Keys;

**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;

**import** org.w3c.dom.Element;

**public** **class** WebdriverWait {

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

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

driver.navigate().to("https://www.flipkart.com/");

// pageLoadTimeout

// Sets the amount of time to wait for a page load to complete before throwing an error. If the timeout is negative, page loads can be indefinite.

driver.manage().timeouts().pageLoadTimeout(20, TimeUnit.***SECONDS***);

// setScriptTimeout

// Sets the amount of time to wait for an asynchronous script to finish execution before throwing an error. If the timeout is negative, then the script will be allowed to run indefinitely.

driver.manage().timeouts().setScriptTimeout(20, TimeUnit.***SECONDS***);

//implicitlyWait

driver.manage().timeouts().implicitlyWait(20, TimeUnit.***SECONDS***);

//Fullscreen the current window if it is not already fullscreen. Its as good as pressing F11 showing no URL bar or menu etc

driver.manage().window().fullscreen();

Thread.*sleep*(2000);

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

driver.findElement(By.*xpath*("//body[@class='fk-modal-visible']")).sendKeys(Keys.***ESCAPE***);

WebDriverWait wait = **new** WebDriverWait(driver, 20);

wait.until(ExpectedConditions.*visibilityOf*(driver.findElement(By.*linkText*("Login & Signup"))));

driver.findElement(By.*linkText*("Login & Signup")).click();

}

}

<https://seleniumjava.com/2016/04/05/the-beginners-guide-to-explicit-waits/>

**Managing tabbed windows in IE, Chrome and Mozilla**

**import** java.awt.AWTException;

**import** java.awt.event.KeyEvent;

**import** java.util.Iterator;

**import** java.util.Set;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.Keys;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.openqa.selenium.interactions.Actions;

**import** com.sun.glass.ui.Robot;

**public** **class** Multiple\_Windows {

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

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

Thread.*sleep*(3000);

// Using below two lines does not add new tab in browser as mentioned in some of

// google search pages

// driver.findElement(By.xpath("//body")).sendKeys(Keys.CONTROL+"t");

// driver.findElement(By.cssSelector("body")).sendKeys(Keys.CONTROL +"t");

// Using below two lines does not add new tab in browser as mentioned in some of

// google search pages

// String selectLinkOpeninNewTab = Keys.chord(Keys.CONTROL,"t");

// driver.findElement(By.linkText("urlLink")).sendKeys(selectLinkOpeninNewTab);

// Using below two lines does not add new tab in browser as mentioned in some of

// google search pages

// Actions a = new Actions(driver);

// a.keyDown(Keys.CONTROL).sendKeys("t").keyUp(Keys.CONTROL).build().perform();

// Robot class is used for Key press and Mouse movement

java.awt.Robot robot1 = **new** java.awt.Robot();

// press Ctrl + t

robot1.keyPress(KeyEvent.***VK\_CONTROL***);

robot1.keyPress(KeyEvent.***VK\_T***);

// Release t followed by control key

robot1.keyRelease(KeyEvent.***VK\_T***);

robot1.keyRelease(KeyEvent.***VK\_CONTROL***);

// Add sleep as it takes time to add a tab in browser

Thread.*sleep*(3000);

// Add another using Robot class

java.awt.Robot robot2 = **new** java.awt.Robot();

robot2.keyPress(KeyEvent.***VK\_CONTROL***);

robot2.keyPress(KeyEvent.***VK\_T***);

robot2.keyRelease(KeyEvent.***VK\_T***);

robot2.keyRelease(KeyEvent.***VK\_CONTROL***);

// Add sleep as it takes time to add a tab in browser

Thread.*sleep*(3000);

// Get all window handles i.e. handles for all available tabs

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

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

**while** (i.hasNext()) {

String windowname = i.next();

// Use to move focus to a particular tab

driver.switchTo().window(windowname);

// Add sleep as it takes time to move focus to another tab in browser

Thread.*sleep*(2000);

driver.navigate().to("http://zeenews.india.com/");

// Add sleep as it may take time to load web site hence window title and other parameters could be blank

Thread.*sleep*(3000);

System.***out***.println("Window handle = " + windowname);

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

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

}

}

}

Another example:

**import** java.util.Iterator;

**import** java.util.Set;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.chrome.ChromeDriver;

**public** **class** Naukri {

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

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

driver.navigate().to("https://www.naukri.com/");

Thread.*sleep*(3000);

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

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

**while** (i.hasNext()) {

String handlename = i.next();

System.***out***.println("Handle name = " + handlename);

driver.switchTo().window(handlename);

Thread.*sleep*(2000);

System.***out***.println("Title = " + driver.getTitle());

System.***out***.println("Link = " + driver.getCurrentUrl());

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

}

}

}

**How to upload file using AutoIt**

**import** java.io.IOException;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** java.io.IOException;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public** **class** Selenium17 {

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

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

driver.navigate().to("http://softwaretestingplace.blogspot.com/2015/10/sample-web-page-to-test.html");

Thread.*sleep*(3000);

driver.findElement(By.*id*("uploadfile")).click();

Thread.*sleep*(3000);

Runtime.*getRuntime*().exec("D:\\Selenium Workspace Oxygen\\Tuition2\\src\\UploadFile.exe");

Thread.*sleep*(3000);

driver.findElement(By.*linkText*("Download Text File")).click();

Runtime.*getRuntime*().exec("D:\\Selenium Workspace Oxygen\\Tuition2\\src\\UploadFile.exe");

}

}

Below is the UploadFile.au3 script present at location D:\\Selenium Workspace Oxygen\\Tuition2\\src\\

WinWaitActive("Open","",20)

Send("D:\File.txt")

Sleep(3000)

Send("{ENTER}")

**Homework:**

Write selenium script to download file from below location:

<http://softwaretestingplace.blogspot.com/2015/10/sample-web-page-to-test.html>

**How to handle dynamic webtables**

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**public** **class** DynamicTables {

**public** **static** **void** main(String[] args) {

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

driver.navigate().to("https://www.sharekhan.com/research/research-for-mf/funds-we-like");

List<WebElement> row\_headers = driver.findElements(By.*xpath*("//table[@datatable='ng']//th"));

ArrayList<String> al\_row\_headers = **new** ArrayList<String>();

String s = **null**;

**for** (**int** i = 0; i < row\_headers.size(); i++) {

s = row\_headers.get(i).getText();

**if** (s.equals("")) {

**continue**;

} **else** {

al\_row\_headers.add(s);

// System.out.println(s);

}

}

System.***out***.println(al\_row\_headers.size());

**for** (**int** i = 0; i < al\_row\_headers.size(); i++) {

System.***out***.println(al\_row\_headers.get(i) + " " + al\_row\_headers.get(i).length());

}

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

System.***out***.println("Scheme Names are:");

List<WebElement> Scheme\_Name = driver.findElements(By.*xpath*("//strong[@class='ng-binding']"));

**for** (**int** i = 0; i < Scheme\_Name.size(); i++) {

System.***out***.println(Scheme\_Name.get(i).getText());

}

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

System.***out***.println("Nav values next to scheme name: ");

List<WebElement> nav1 = driver.findElements(

By.*xpath*("//table[@datatable='ng']/tbody//td//span//span[@class='half\_grid text-right ng-binding']"));

**for** (**int** i = 0; i < nav1.size(); i++) {

System.***out***.println(nav1.get(i).getText());

}

System.***out***.println("All rows");

List<WebElement> webtable\_rows = driver.findElements(By.*xpath*("//tr[contains(@class,'ng-scope')]/td"));

**for** (**int** i = 0; i < webtable\_rows.size(); i++) {

**if** ((i % 9) == 0) {

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

}

System.***out***.print(webtable\_rows.get(i).getText() + " ");

}

}

}