Synchronization, HardWait, ImplicitWait

This is the next tutorial in selenium-java series. Please go through the previous tutorials before you start this one. In the last tutorial, we learned about calendars. In this tutorial we will learn how to handle synchronization!

What you will Learn:

1. What is synchronization
2. Hard wait
3. Implicit wait

What is synchronization

Let us consider a typical flight search scenario. To search for a flight, what do we do? We go to a website, select departure city, select arrival city, enter a date we would like to travel, hit search. The website starts searching and the page loads after sometime (may be 3 seconds or more, depending) showing the search results. See figures 1,2 & 3.

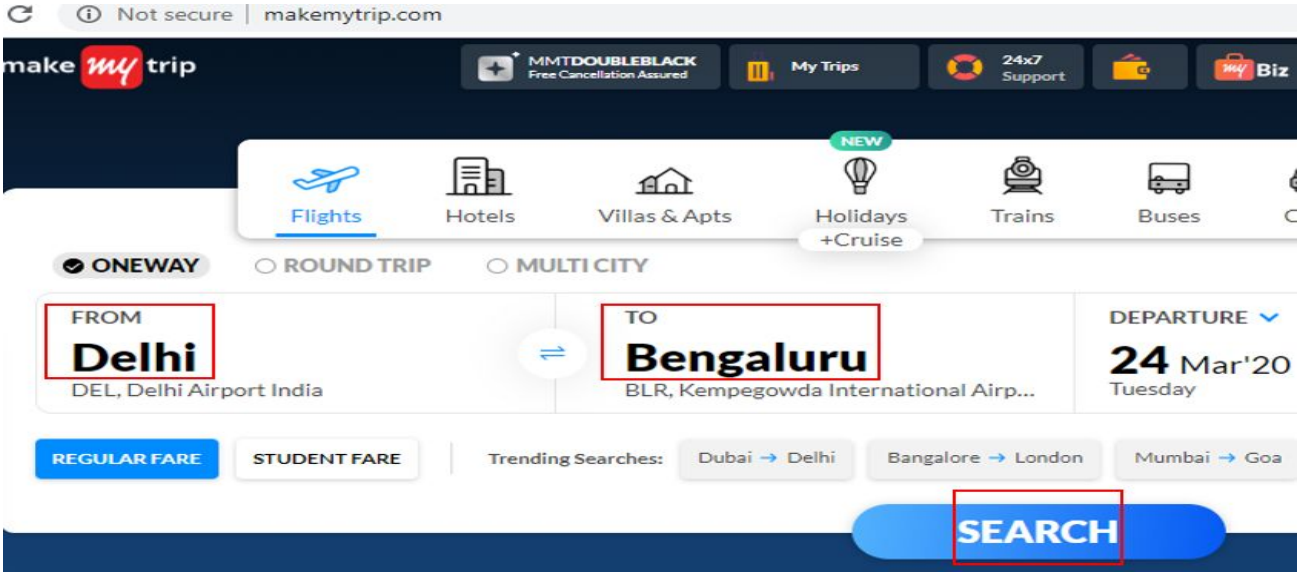


Figure 1

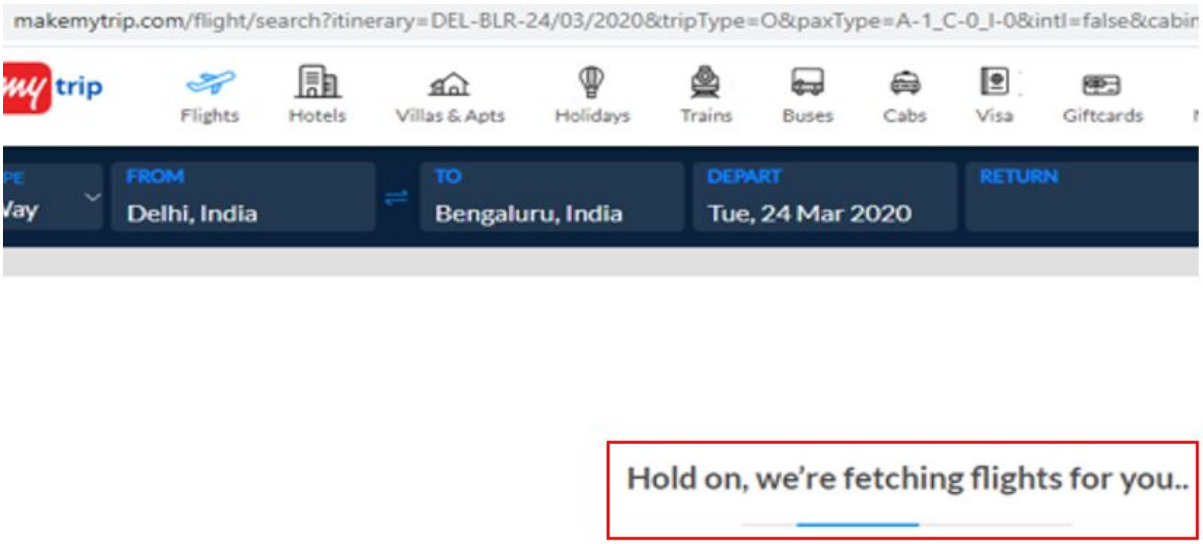


Figure 2

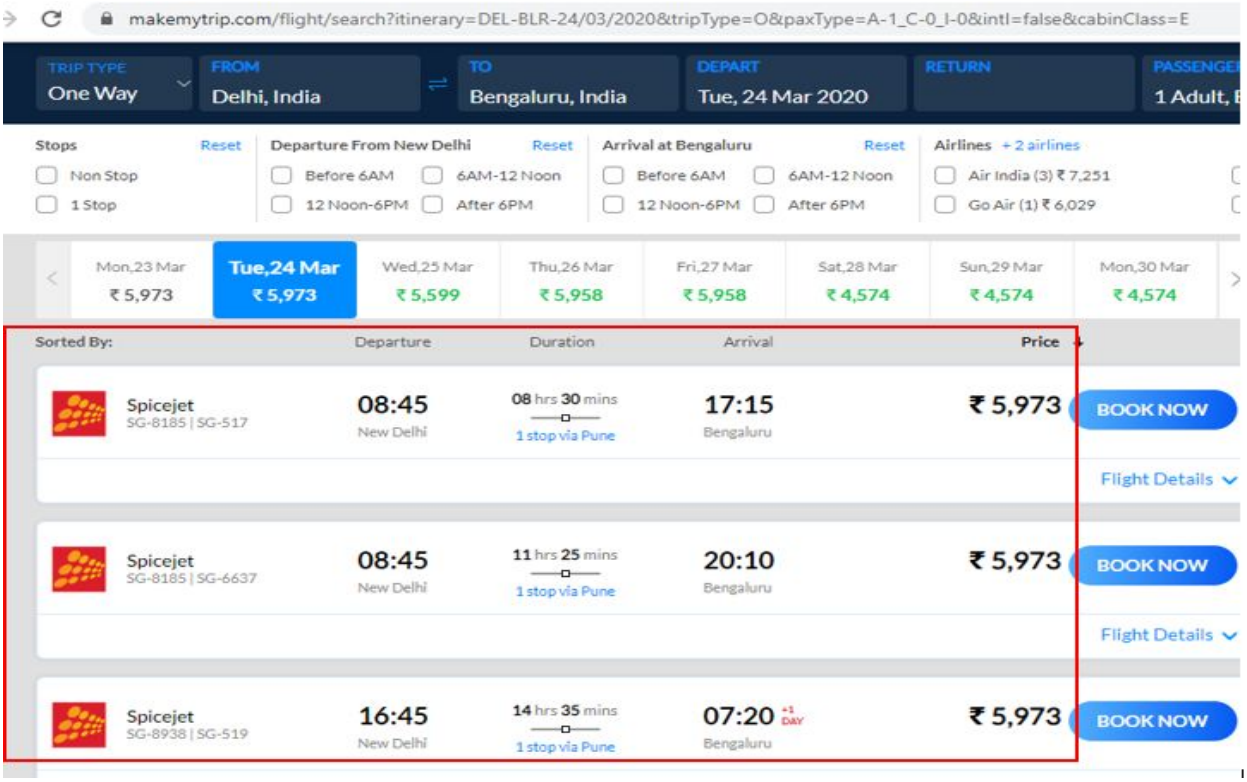


Figure 3

In real time, there will be some delay in displaying the search results (see figure 2). The tricky part here is that, you do not know in advance, how much time the website would take to display the search results. So the selenium script should be designed in such a way that it waits till the search results page (figure 3) is completely loaded. This is called synchronization. If you don't tell selenium to wait for a certain time, it might try to click 'BOOK NOW' element immediately (figure 3) & script might fail, saying "Element not found exception...".

There are 4 ways to achieve this synchronization: Hard wait, Implicit wait, explicit wait and fluent wait.

Hard wait

Before we study about 'Hard wait', let us perform these steps manually:

---Navigate to https://alaskatrips.poweredbygps.com/g/pt/hotels

---In the 'Leaving from' field, enter **nyc**. You will see a dynamic dropdown with options shown

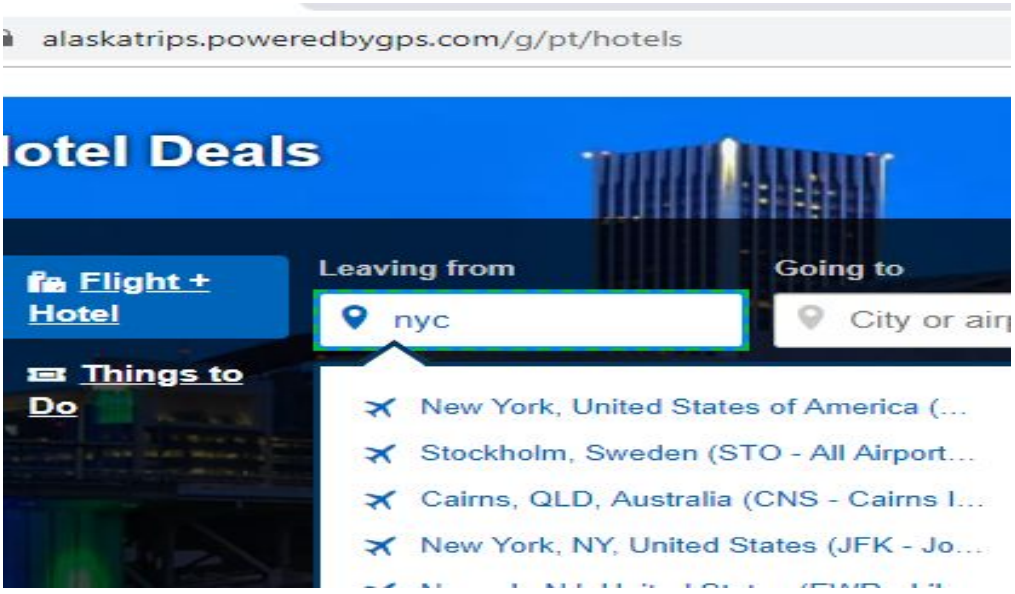


Figure 4

Perform one arrow key down, you will notice that a place gets selected

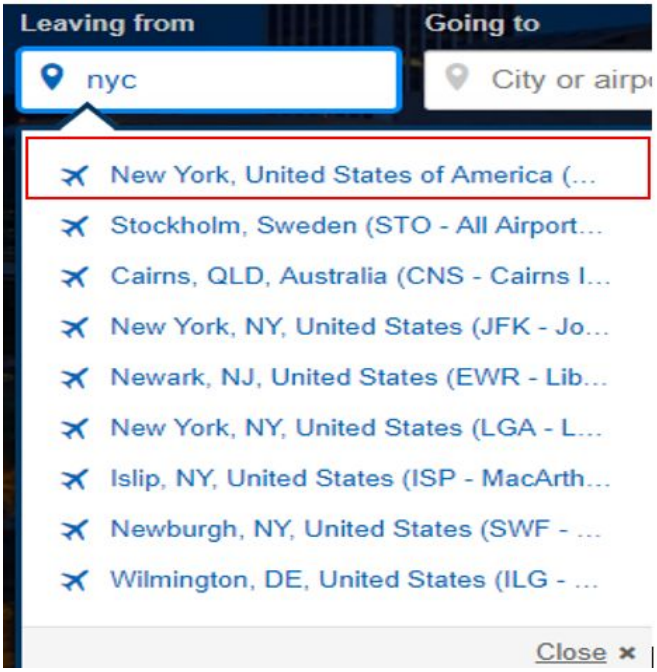


Figure 5

Hit 'Enter', the 'Leaving from field will get populated with the text of that place

Figure 6

Let us try to automate these manual steps that we have performed. If we inspect 'Leaving from' field, we notice that it is uniquely identified by 'id'

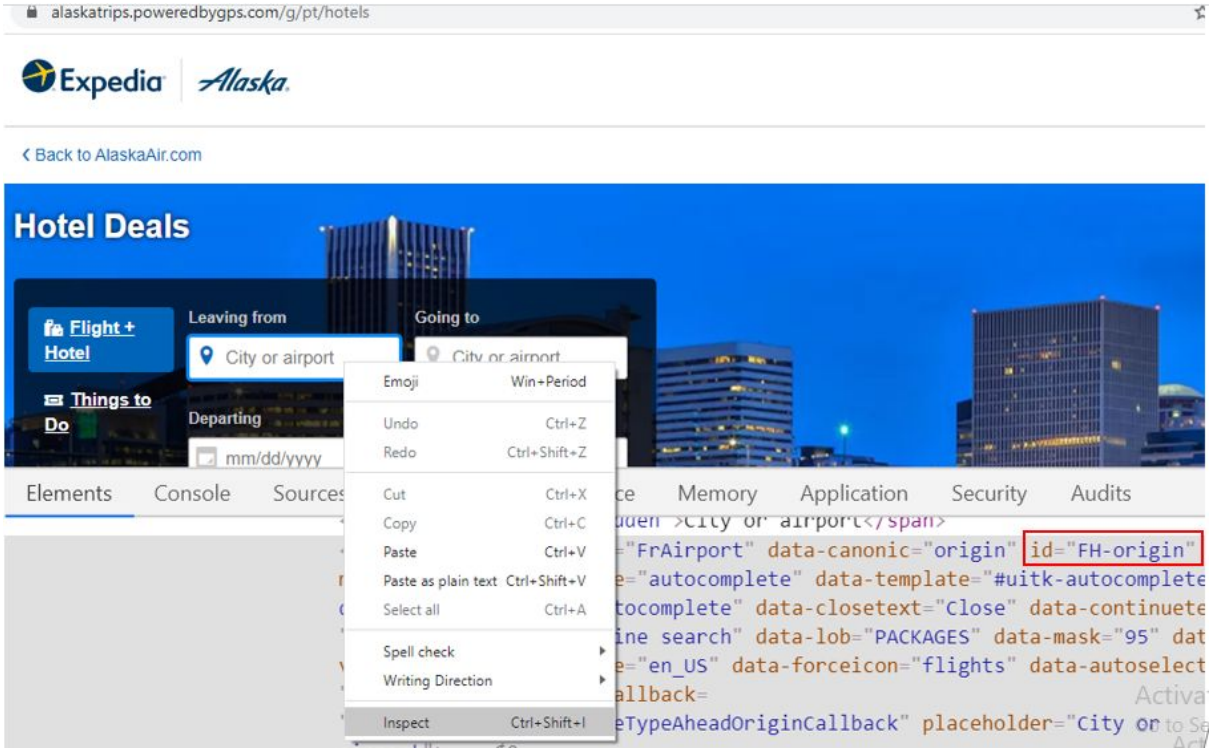


Figure 7

Create a new class 'HardWait' having main method

```
10    public static void main(String[] args) {
11        System.setProperty("webdriver.chrome.driver", "C:\\Users\\DELL\\Desktop\\TI
12
13        WebDriver driver = null;
14
15        driver = new ChromeDriver();
16
17        driver.navigate().to("https://alaskatrips.poweredbygps.com/g/pt/hotels");
18        driver.findElement(By.id("FH-origin")).sendKeys("nyc");
19        driver.findElement(By.id("FH-origin")).sendKeys(Keys.ARROW_DOWN);
20        driver.findElement(By.id("FH-origin")).sendKeys(Keys.ENTER);
```

Figure 8

Run this script, notice the browser page that selenium opens (figure 9)

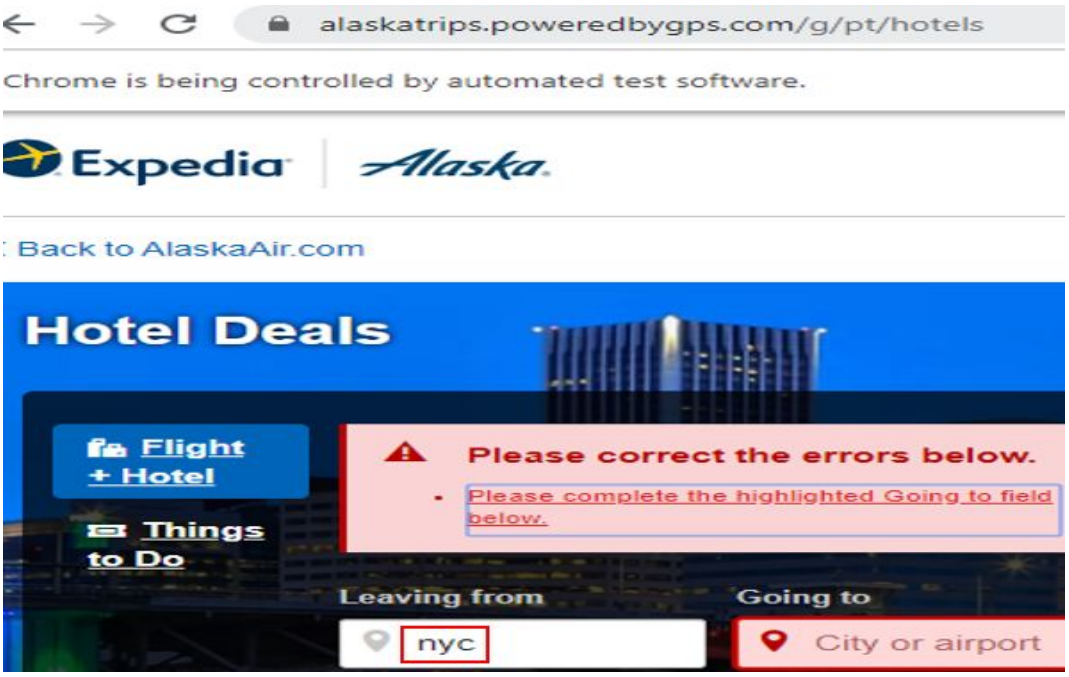


Figure 9

You will notice that the script enters text 'nyc' but fails at step 19. The reason being, the script tries to perform arrow down operation but it does not find any dynamic dropdown list & hence fails. Let us fix this synchronization issue using hard wait. Let us now add a hard wait of 2 seconds after sending the text 'nyc', see line#19 below

```
 8 public class HardWait {
 9
10     public static void main(String[] args) throws InterruptedException {
11         System.setProperty("webdriver.chrome.driver", "C:\\Users\\DELL\\Desktop\\
12
13         WebDriver driver = null;
14
15         driver = new ChromeDriver();
16
17         driver.navigate().to("https://alaskatrips.poweredbygps.com/g/pt/hotels");
18         driver.findElement(By.id("FH-origin")).sendKeys("nyc");
19         Thread.sleep(2000);
20         driver.findElement(By.id("FH-origin")).sendKeys(Keys.ARROW_DOWN);
21         driver.findElement(By.id("FH-origin")).sendKeys(Keys.ENTER);
```

Figure 10

Run the script, now you would see that the 'key down' & 'Enter' operations are performed, the city gets selected from dropdown, the text appears in the 'Leaving from' field, see below. So basically, after the script populates 'nyc' in step#18, it waits for 2 second. Due to this wait time, the dropdown list comes up during runtime and the script is able to perform keydown operation and select a value from dropdown list. This is how the sync issue got resolved in this case.

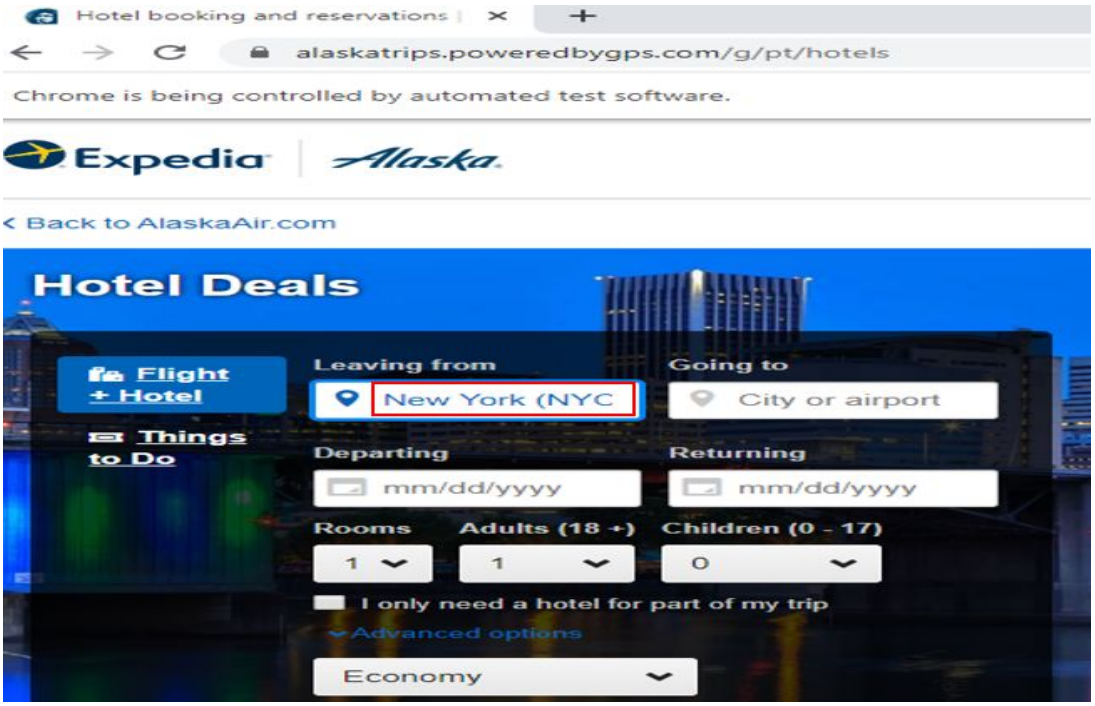


Figure 11a

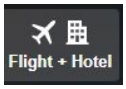
Please note that, we should not use hard wait to resolve all the synchronization issues. We should use it only if it's necessary. The reason being, hard waits will increase the total execution time of your scripts. So, in the above scenario, even if the dropdown list appears in 1 second, the script will still wait for 2 seconds, since 2 seconds is the hard wait time that we have defined in the script.

Implicit wait

We know that, each web page takes some time to load. In the case of implicit wait, we define a 'global' wait timeout. So we will ask selenium to wait for 'n' number of seconds before it throws any error. So, if the 'driver' doesn't find the desired element, we will be instructing the driver not to throw any error immediately. Instead, the driver should wait for the time mentioned in global timeout & then try to perform an operation on the element. If the driver still does not find an element (after waiting for time defined in implicit timeout), than throw an exception.

Let us suppose that the implicit wait timeout that we have defined in the script is 5 secs. Now, let's suppose that the search results get displayed in 2 secs. In this case, the script will not wait for entire 5 secs (unlike hard wait). If the search results get displayed in 2 seconds, the script will move ahead and perform the remaining operations. In this respect, implicit wait is better than hard wait.

Our use case that we will automate:
---Navigate to https://www.expedia.co.in/

---click Flight+Hotel 

. The reason we have to click this is because, when selenium launches the site, it lands on 'Hotels' page by default (see figure 11b). However when we navigate to the site manually, we land up at 'Flight + Hotel' page by default

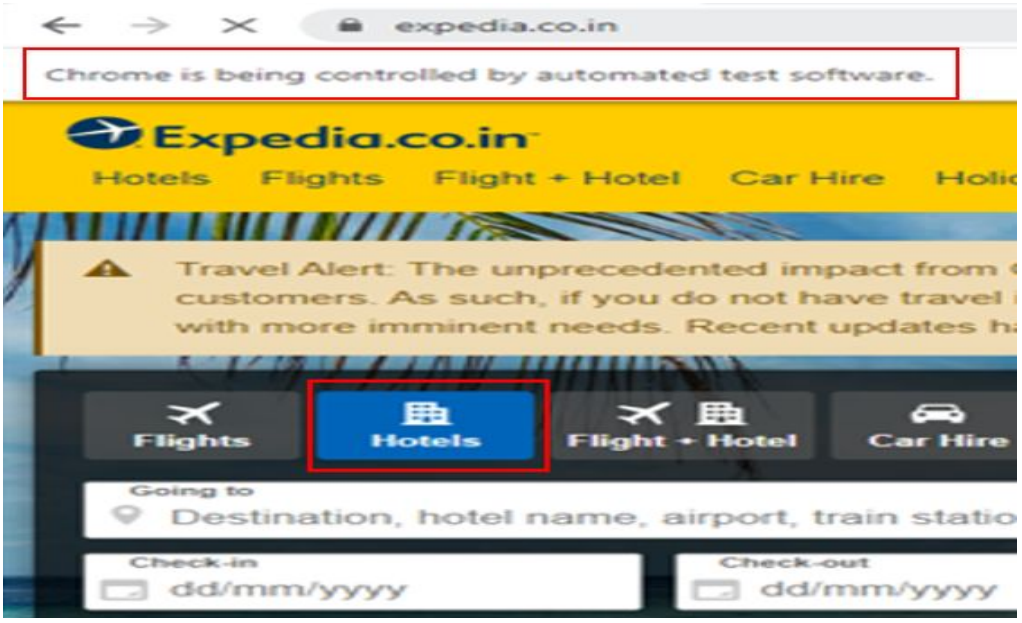


Figure 11b: Selenium behaviour, lands at 'Hotels' page

---Enter 'c' in the Origin field, select first option from dropdown
---Enter 'on' in the destination field, select first option from dropdown
---Enter departing date

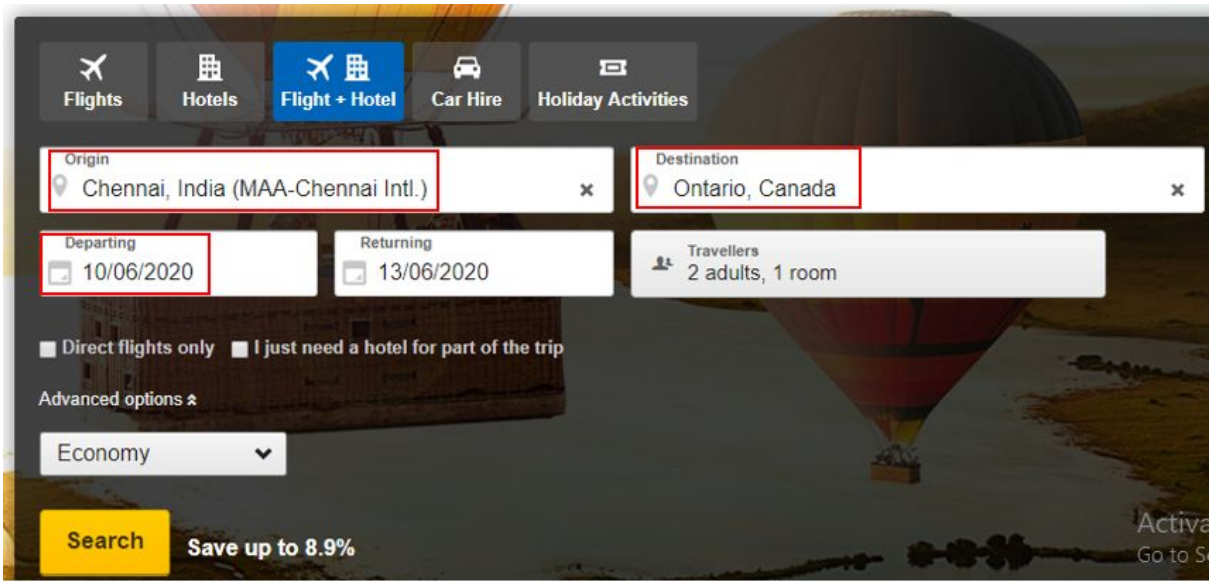


Figure 12

---Click Search

Figure 13

---Lot of search results would come up as shown below

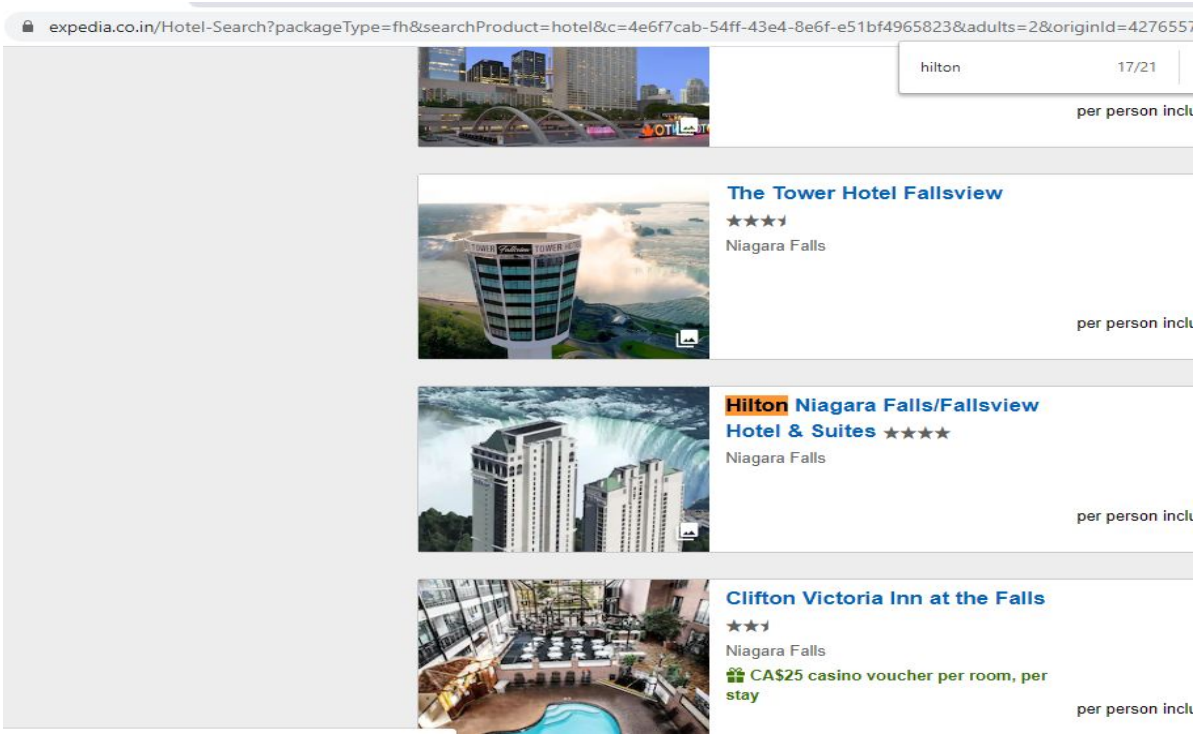


Figure 14

--- Our task is to search for the word ‘Hilton’ and click the hotel link from the search results. The hotel link would open in a new tab window

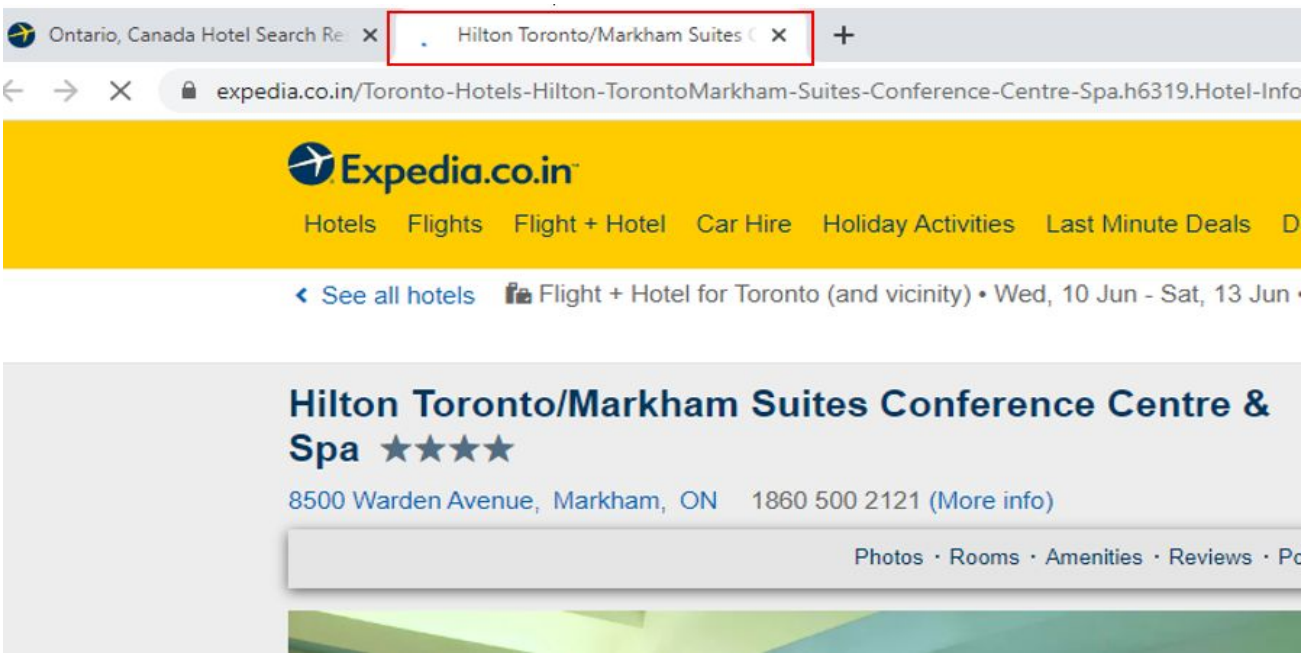


Figure 15

Let us automate these steps. Inspect 'Flight + Hotel' and create custom xpath

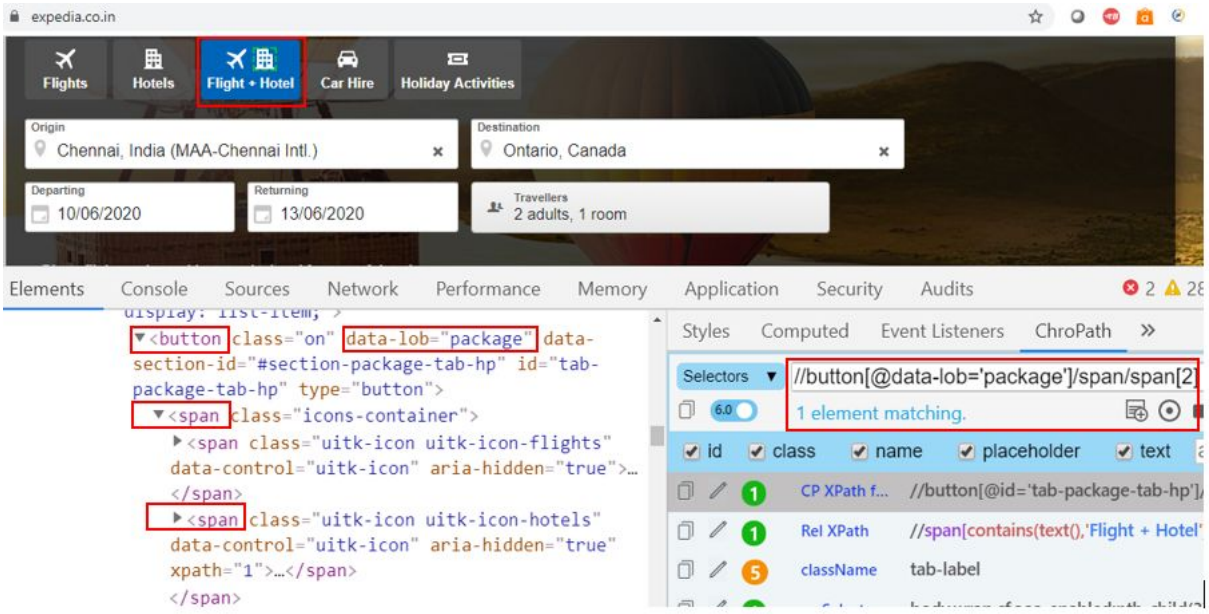


Figure 16

Similarly, the ‘origin’ city xpath can be identified by ‘id’: package-origin-hp-package

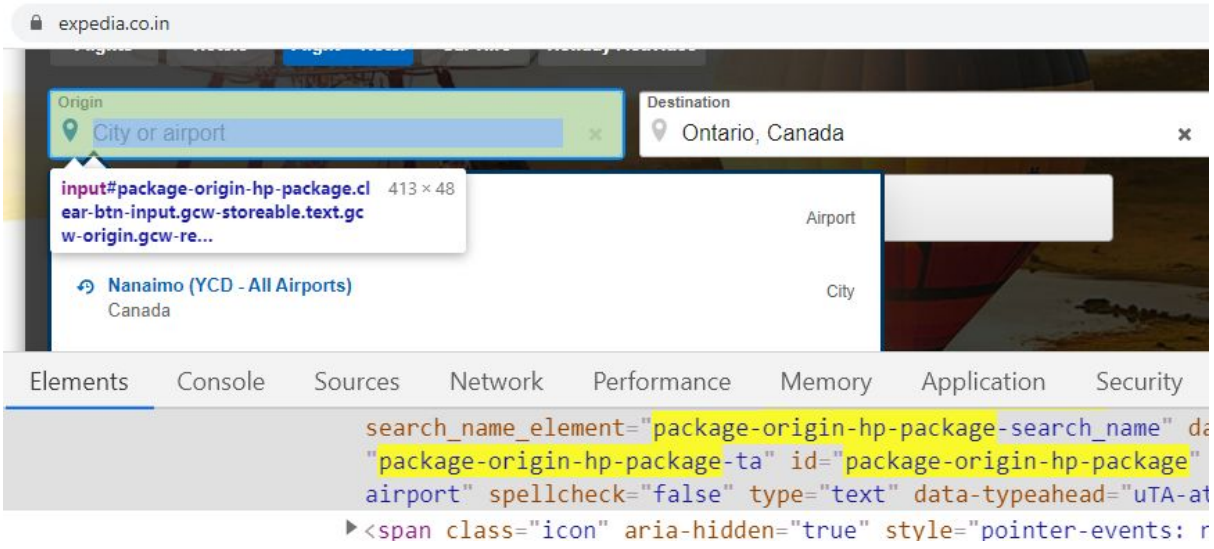


Figure 17

Similarly, the ‘destination’ city xpath can be identified by ‘id’: package-destination-hp-package

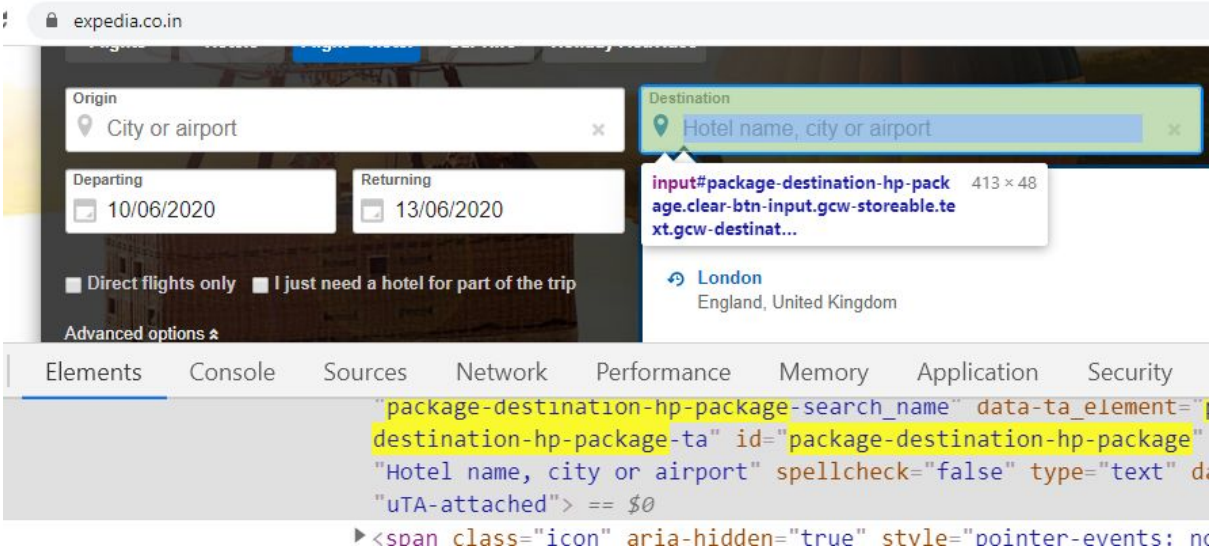


Figure 18

Similarly, the ‘Departing’ xpath can be identified by ‘id’: package-departing-hp-package

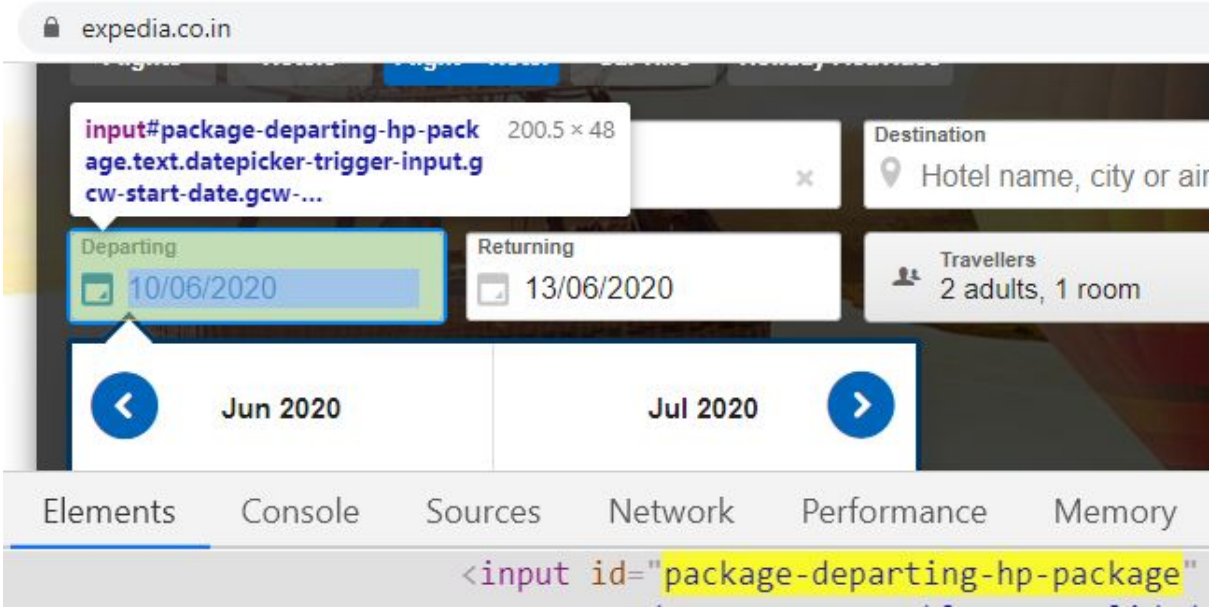


Figure 19

We will use these xpaths in our new class ‘ImplictWait’. First we will see the results without implementing implicit wait timeout.

Change line 41 (since this hotel might not exist) & the departure date at line 38 (if needed)

```
21          driver.navigate().to("https://www.expedia.co.in/");
22
23          driver.findElement(By.xpath("//button[@data-lob='package']/span/span[2]")).click();
24
25          driver.findElement(By.id("package-origin-hp-package")).sendKeys("c");
26          Thread.sleep(2000);
27          driver.findElement(By.id("package-origin-hp-package")).sendKeys(Keys.ARROW_DOWN);
28          driver.findElement(By.id("package-origin-hp-package")).sendKeys(Keys.ENTER);
29          driver.findElement(By.id("package-origin-hp-package")).sendKeys(Keys.TAB);
30          driver.findElement(By.id("package-origin-hp-package")).sendKeys(Keys.TAB);
31          Thread.sleep(2000);
32
33          driver.findElement(By.id("package-destination-hp-package")).sendKeys("on");
34          Thread.sleep(2000);
35          driver.findElement(By.id("package-destination-hp-package")).sendKeys(Keys.ARROW_DOWN);
36          driver.findElement(By.id("package-destination-hp-package")).sendKeys(Keys.ENTER);
37          Thread.sleep(1000);
38          driver.findElement(By.id("package-departing-hp-package")).sendKeys("10/06/2020");
39          driver.findElement(By.id("package-departing-hp-package")).sendKeys(Keys.ENTER);
40
41          driver.findElement(By.xpath("//a[contains(@href,'Hilton')]")).click();
42      }
```

Figure 20

Run the script, see figure 21. You can see an error at the background. So even though the search is still in progress, the script tries to execute line 41 immediately. So selenium tries to click the hotel link immediately even though search result page has not loaded. This leads to failure. This is where the implicit wait would come into picture.

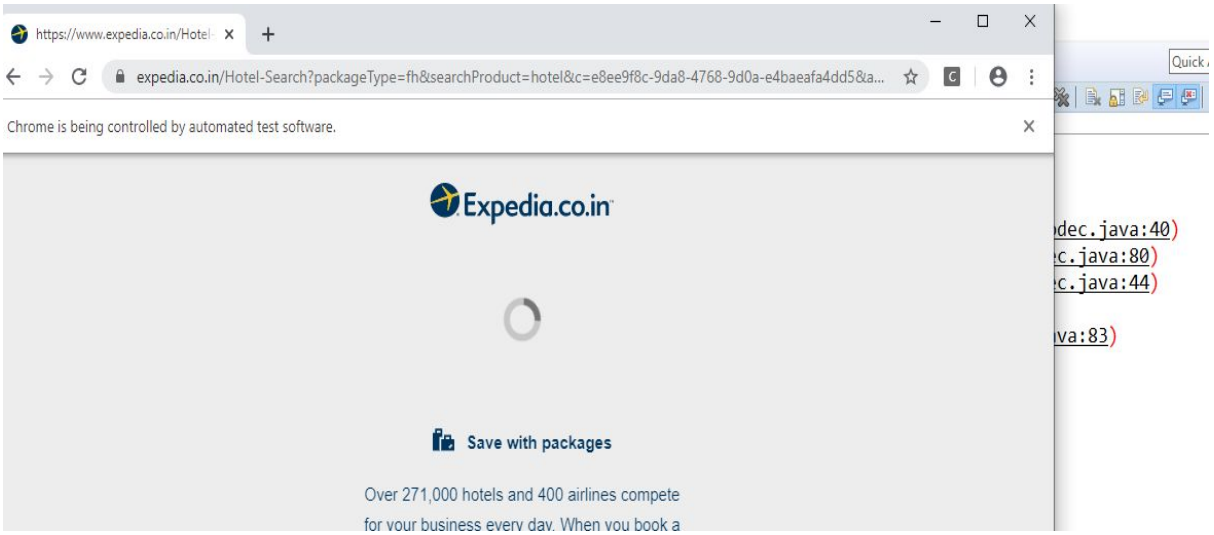


Figure 21

Notice that NoSuchElement exception is thrown

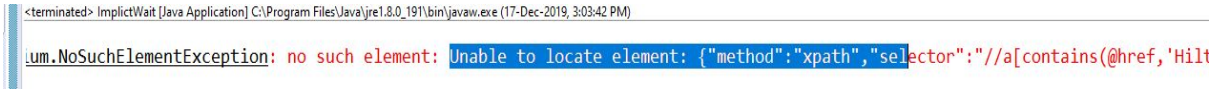


Figure 22

Using hard wait of 15 seconds at line#40 (see below figure) is a bad practice. Even though the script might pass this time but it's a bad design. We should never use more than 2 or 3 seconds of hard wait.

```
21          driver.navigate().to("https://www.expedia.co.in/");
22
23          driver.findElement(By.xpath("//button[@data-lob='package']/span/span[2]")).click();
24
25          driver.findElement(By.id("package-origin-hp-package")).sendKeys("c");
26          Thread.sleep(2000);
27          driver.findElement(By.id("package-origin-hp-package")).sendKeys(Keys.ARROW_DOWN);
28          driver.findElement(By.id("package-origin-hp-package")).sendKeys(Keys.ENTER);
29          driver.findElement(By.id("package-origin-hp-package")).sendKeys(Keys.TAB);
30          driver.findElement(By.id("package-origin-hp-package")).sendKeys(Keys.TAB);
31          Thread.sleep(2000);
32
33          driver.findElement(By.id("package-destination-hp-package")).sendKeys("on");
34          Thread.sleep(2000);
35          driver.findElement(By.id("package-destination-hp-package")).sendKeys(Keys.ARROW_DOWN);
36          driver.findElement(By.id("package-destination-hp-package")).sendKeys(Keys.ENTER);
37          Thread.sleep(1000);
38          driver.findElement(By.id("package-departing-hp-package")).sendKeys("10/06/2020");
39          driver.findElement(By.id("package-departing-hp-package")).sendKeys(Keys.ENTER);
40          Thread.sleep(15000); //BAD PRACTICE
41          driver.findElement(By.xpath("//a[contains(@href,'Hilton')]")).click();
```

Figure 23

Lets now add implicit wait of 25 seconds globally (immediately after driver definition), line#19. This is the standard syntax of implicit wait. Also note that, we have used hard waits as well at other places that cannot be tacked by implicit waits.

```
17          driver = new ChromeDriver();
18
19          driver.manage().timeouts().implicitlyWait(25, TimeUnit.SECONDS);
20
21          driver.navigate().to("https://www.expedia.co.in/");
22
23          driver.findElement(By.xpath("//button[@data-lob='package']/span/span[2]")).click();
24
25          driver.findElement(By.id("package-origin-hp-package")).sendKeys("c");
26          Thread.sleep(2000);
27          driver.findElement(By.id("package-origin-hp-package")).sendKeys(Keys.ARROW_DOWN);
28          driver.findElement(By.id("package-origin-hp-package")).sendKeys(Keys.ENTER);
29          driver.findElement(By.id("package-origin-hp-package")).sendKeys(Keys.TAB);
30          driver.findElement(By.id("package-origin-hp-package")).sendKeys(Keys.TAB);
31          Thread.sleep(2000);
32
33          driver.findElement(By.id("package-destination-hp-package")).sendKeys("on");
34          Thread.sleep(2000);
35          driver.findElement(By.id("package-destination-hp-package")).sendKeys(Keys.ARROW_DOWN);
36          driver.findElement(By.id("package-destination-hp-package")).sendKeys(Keys.ENTER);
37          Thread.sleep(1000);
38          driver.findElement(By.id("package-departing-hp-package")).sendKeys("10/06/2020");
39          driver.findElement(By.id("package-departing-hp-package")).sendKeys(Keys.ENTER);
40
41          driver.findElement(By.xpath("//a[contains(@href,'Hilton')]")).click();
```

Figure 24

Run script, notice this time that no exception is thrown while the search is in progress, see below

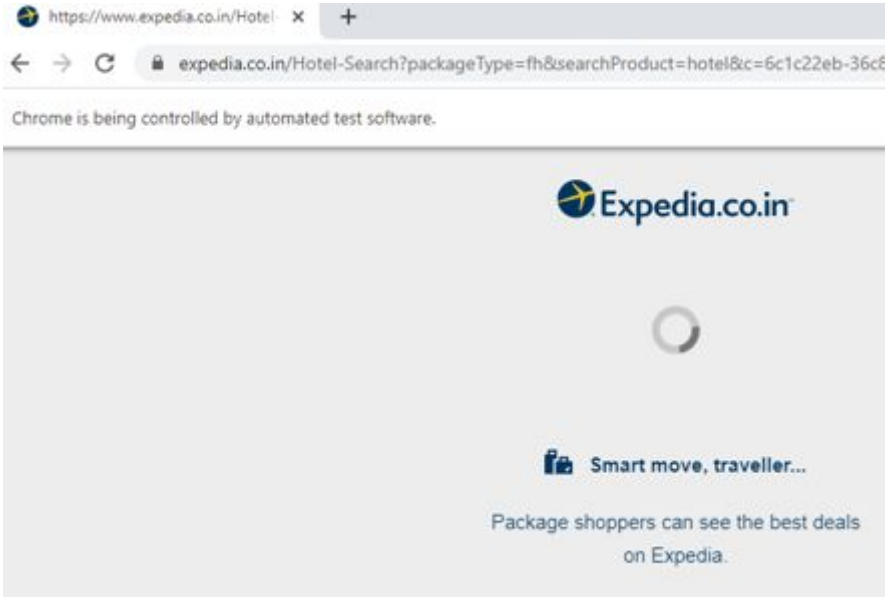


Figure 25

The script now waits for the results page to load & it than clicks the hotel link as soon as it finds it, see below. If the search page loads prior to 25 seconds, than the script performs the next operation, it does not wait for entire 25 secs

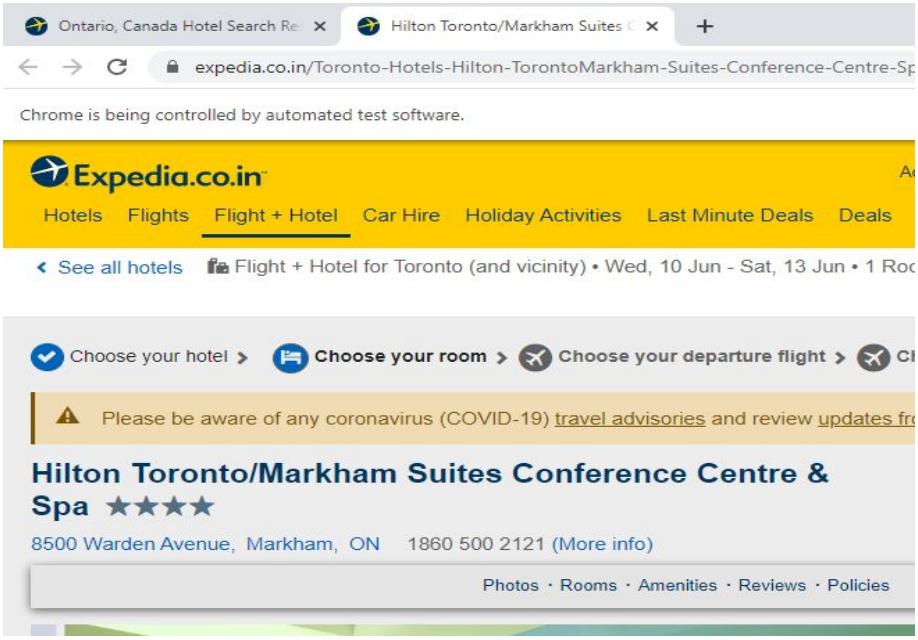


Figure 26

No error is thrown in the console

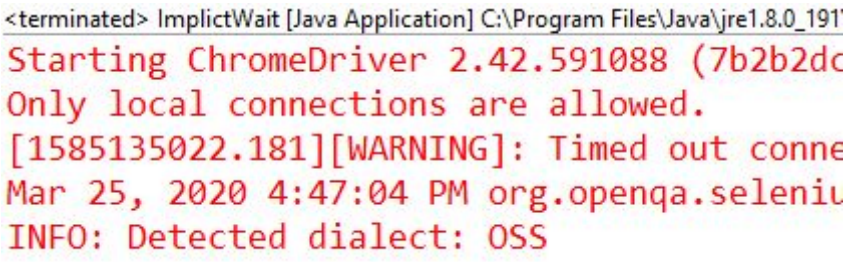


Figure 27

So this is how implicit wait works.

Note: We should use a mix of hard wait, implicit and explicit waits in our scripts.

In our next tutorial, we will learn about explicit and fluent waits. Thank you for reading!