

## ExplicitWait

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 hard and implicit wait. In this tutorial we will learn about explicit wait!

### **What you will Learn:**

1. Disadvantage of implicit wait
2. Explicit wait introduction
3. Explicit wait in detail
4. Assignment
5. Code snippet

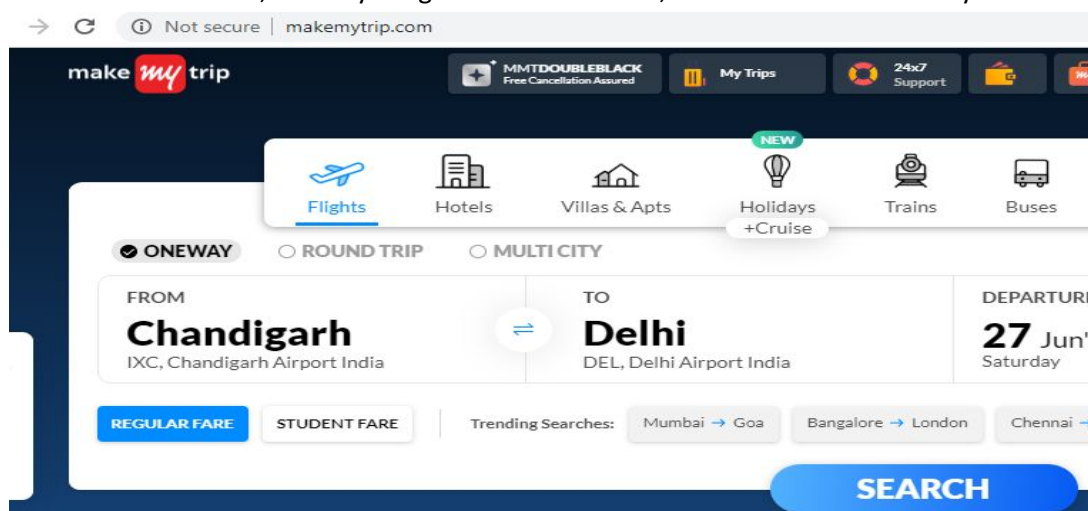
### **Disadvantage of implicit wait**

Let's say, we want to search hotels in 'New York'. Let's suppose only 50 results are displayed within 5 seconds. In this case, implicit wait of 10 seconds will be a PASS. Now, if suppose, we want to search hotels in entire US. Let's say 10 thousand results are displayed within 15 seconds (viz it takes 15 seconds to load search results). In the latter case, implicit wait of 10 seconds will FAIL. Now you might think of increasing implicit time to 15 secs to handle latter case. This will make your script pass, however this is a disadvantage as well. Let's say QA manager has given you a performance requirement saying that the search results should be displayed within 5 secs (irrespective how many hotels we are searching), then you will not be able to catch this performance issue if you increase the implicit wait time. It might also be that after few months, even a single city search starts taking 7 seconds to load, you will not be able to catch this performance issue since you increased the implicit wait and hence your script passed. So in summary, if you increase implicit wait time, you can't catch the performance issue. But if you decrease the implicit time to 5 secs, than 15 secs search result tests will fail.

To handle these kind of scenarios, explicit wait comes into picture. Using explicit wait, you can target specific element(s).

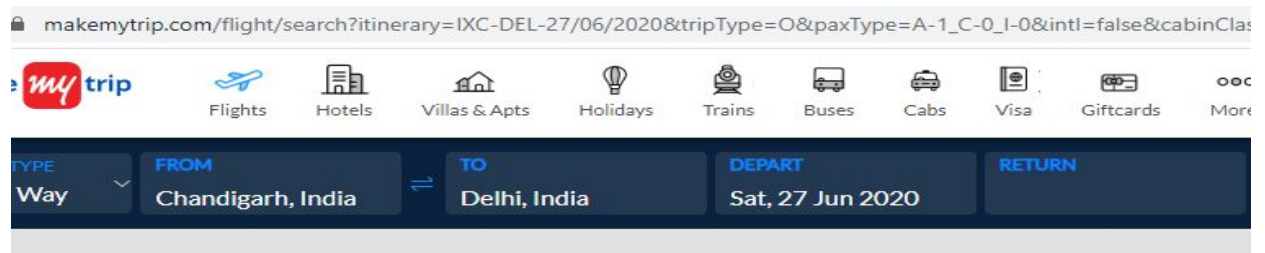
### **Explicit wait introduction**

To understand better, let's say we go to below website, enter 'From' and 'To' city



**Figure 1A**

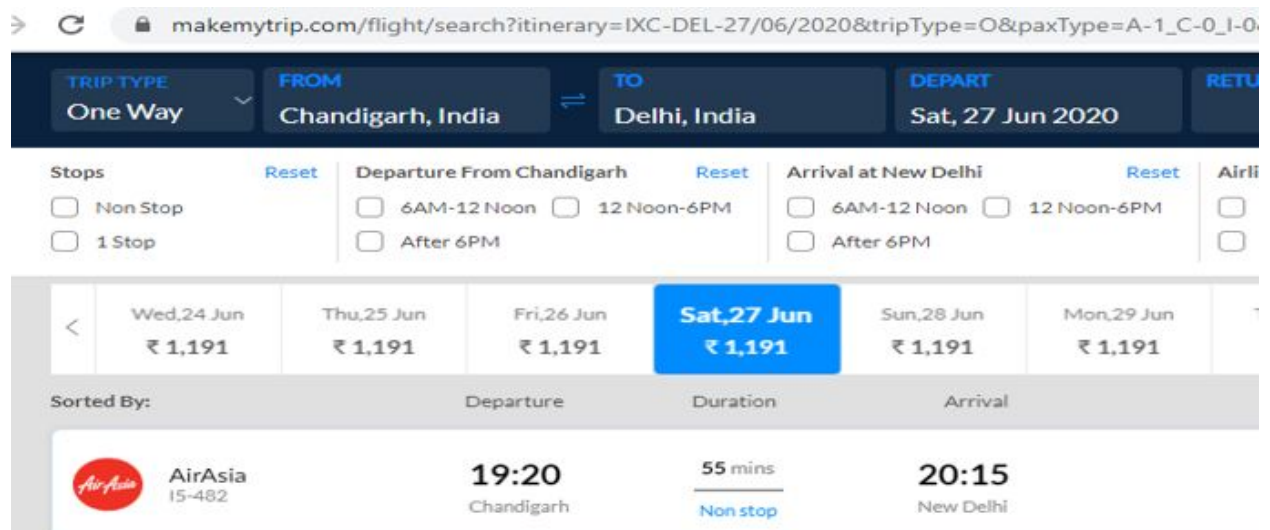
Click Search, below intermediate window comes up



Hold on, we're fetching flights for you..

**Figure 1B**

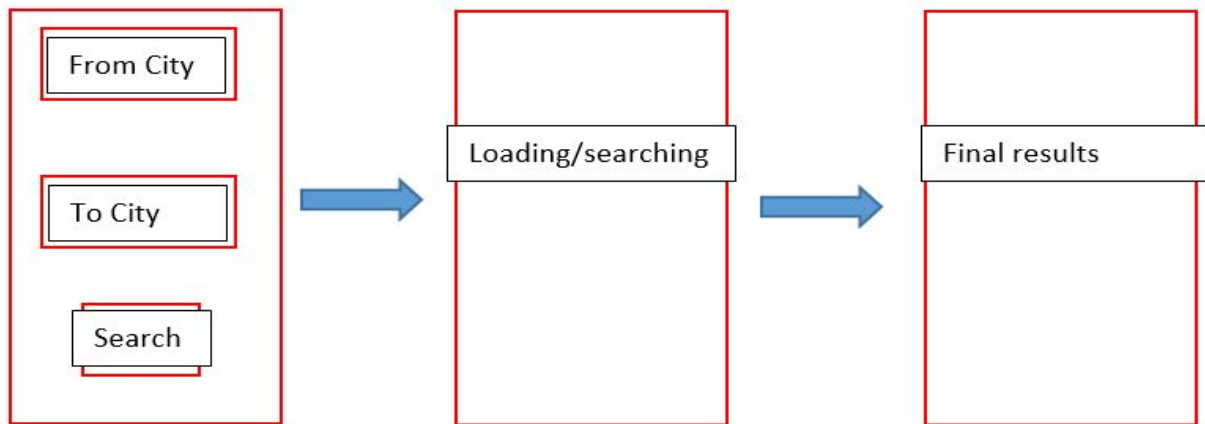
The below search results comes up



**Figure 1C**

Sometimes, the search results (above window) will take a lot of time to come up. You might see the previous 'Hold on, we're fetching flights for you' intermediate window for a longer (unknown) time. So basically, you have to wait dynamically for whatever you are searching on a webpage.

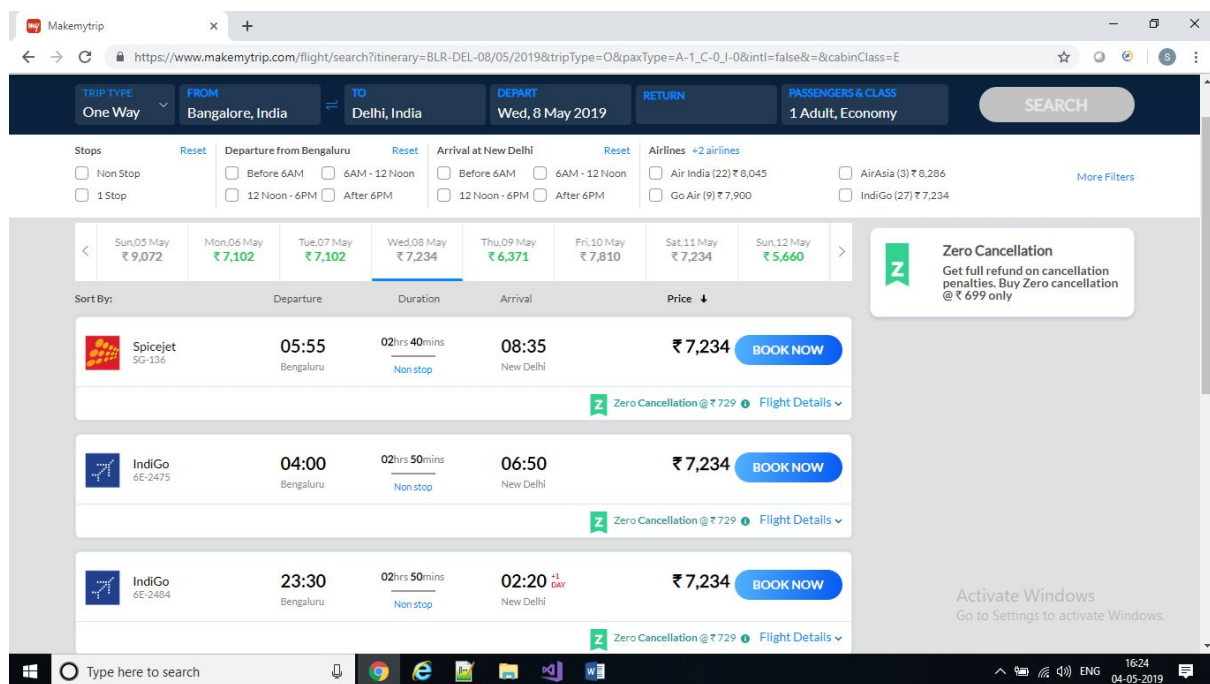
See below, in the left hand box, we enter 2 cities and click search, the middle box is where we saw an intermediate window 'searching for best fares', the rightmost window is the final results



**Figure 1D**

So the implicit wait that you have setup for 20 seconds might not work for the intermediate box 'Hold on, we're fetching flights for you'. So we use explicit wait here.

So we can keep a condition that, till the time we don't get 'BOOK NOW' button, we will have to wait for that much time. So we can tell selenium to explicitly wait for 30 seconds for the 'BOOK NOW' button to become visible instead of 20 seconds implicit wait



**Figure 1E**

### Explicit wait in detail

We have already seen the implicit wait being set to 20 seconds for findElement. This is the default waiting time for which selenium waits to find the objects on the page. Now whenever you write driver.findElement() or driver.findElements(), this would be the default timeout. After 20 seconds it will throw an error if the element is not found. Now sometimes you would want to wait extra for a certain element. This is done using explicit wait.

Explicit wait is NOT a global timeout. Explicit wait is applied to a specific element on a webpage. So if there are 100 elements on a page and if there is a single element that takes a long time to load, we

can make use of explicit wait for that specific element. We should not change the implicit global timeout just for a single element.

Create a new class & copy/paste the code from implicit wait class (that we created in previous article). Comment the implicit wait timeout line#19, see below

```
10 public class ExplicitWait {
11
12     public static void main(String[] args) throws InterruptedException {
13         System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\DELL\\Des
14
15         WebDriver driver = null;
16
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/spa
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(Key
28         driver.findElement(By.id("package-origin-hp-package")).sendKeys(Key
29         driver.findElement(By.id("package-origin-hp-package")).sendKeys(Key
30         driver.findElement(By.id("package-origin-hp-package")).sendKeys(Key
31         Thread.sleep(2000);
32
33         driver.findElement(By.id("package-destination-hp-package")).sendKeys
34         Thread.sleep(2000);
35         driver.findElement(By.id("package-destination-hp-package")).sendKeys
```

**Figure 1F**

Explicit wait is implemented by an inbuilt class '**WebDriverWait**'

Now just before line 42 (see below), we will define a '**WebDriverWait**' class having object reference '**d**'. This reference '**d**' will wait max up-to 25 seconds for an element, after that it will timeout



```

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 WebDriverWait d = new WebDriverWait(driver, 25);
42 driver.findElement(By.xpath("//a[contains(@href, 'Hilton')]")).click();

```

**Figure 2**

We can now use 'until' method and use various conditions on this reference 'd', see below

```

29 driver.findElement(By.id("package-origin-hp-package")).sendKeys("c");
30 Thread.sleep(2000);
31
32 driver.findElement(By.id("package-destination-hp-package")).sendKeys("on");
33 Thread.sleep(2000);
34 driver.findElement(By.id("package-departing-hp-package")).sendKeys("10/06/2020");
35 driver.findElement(By.id("package-departing-hp-package")).sendKeys(Keys.ENTER);
36 Thread.sleep(1000);
37 driver.findElement(By.id("package-departing-hp-package")).sendKeys(Keys.ENTER);
38 driver.findElement(By.id("package-departing-hp-package")).sendKeys(Keys.ENTER);
39
40 WebDriverWait d = new WebDriverWait(driver, 25);
41 d.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//a[contains(@href, 'Hilton')]")));
42 driver.findElement(By.xpath("//a[contains(@href, 'Hilton')]")).click();

```

**Figure 3**

We can see above that there are lot of 'visibility' methods that we can use.

So what we are saying in line#42 is that, *wait until 25seconds for the element to be visible*. Once it is visible, then only click it (line 44). Thus, the reference 'd' only knows about the element located by xpath mentioned in line#44, it does not have knowledge about other elements

```

41 WebDriverWait d = new WebDriverWait(driver, 25);
42 d.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//a[contains(@href, 'Hilton')]")));
43
44 driver.findElement(By.xpath("//a[contains(@href, 'Hilton')]")).click();

```

**Figure 4**

Run the script, the hotel link gets clicked (even though we have disabled implicit wait timeout, there was no error)

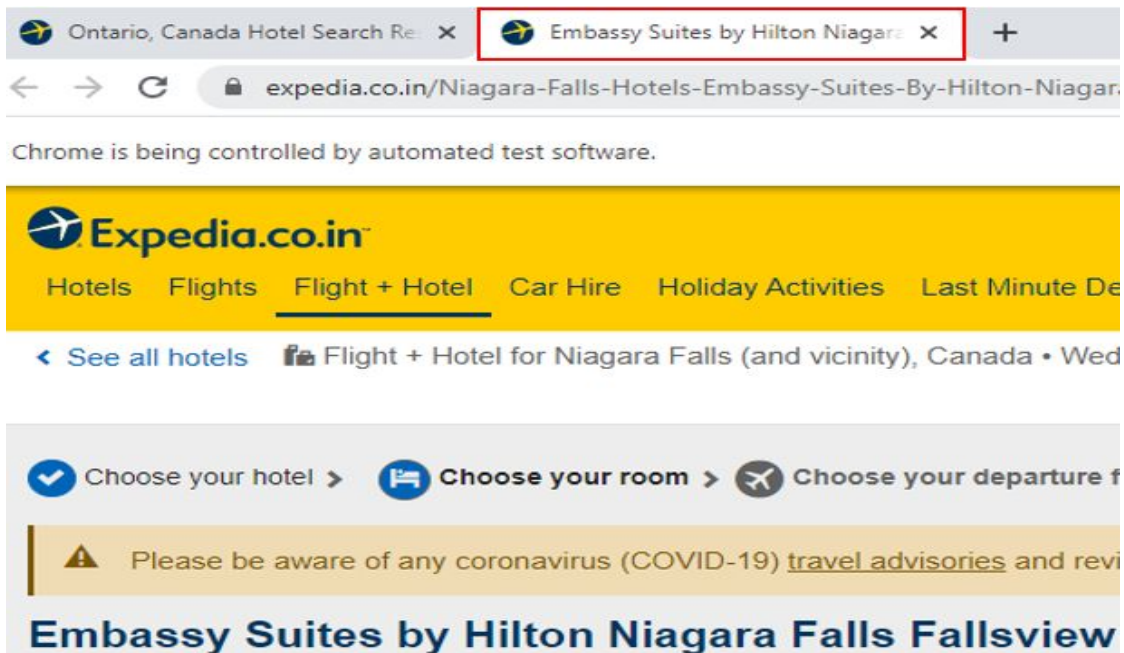


Figure 5

If you search for above hotel keyword 'Hilton', you would notice that it is present on search results page#2, see below

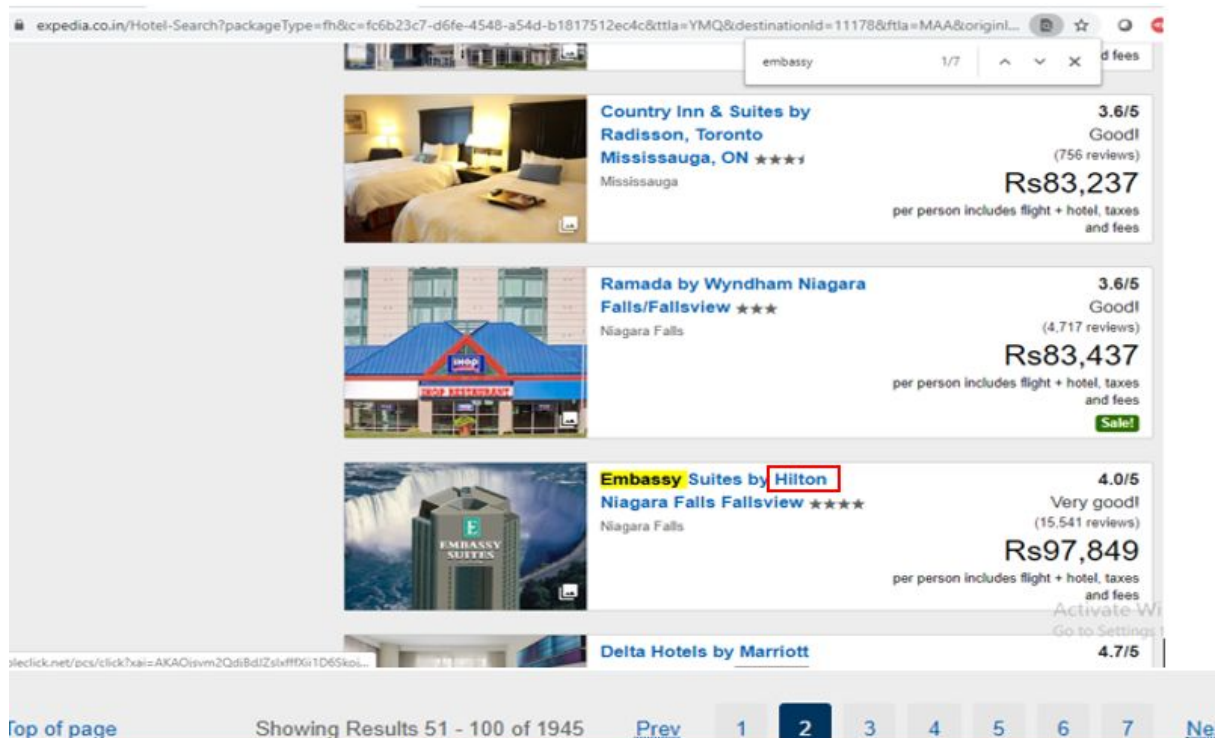
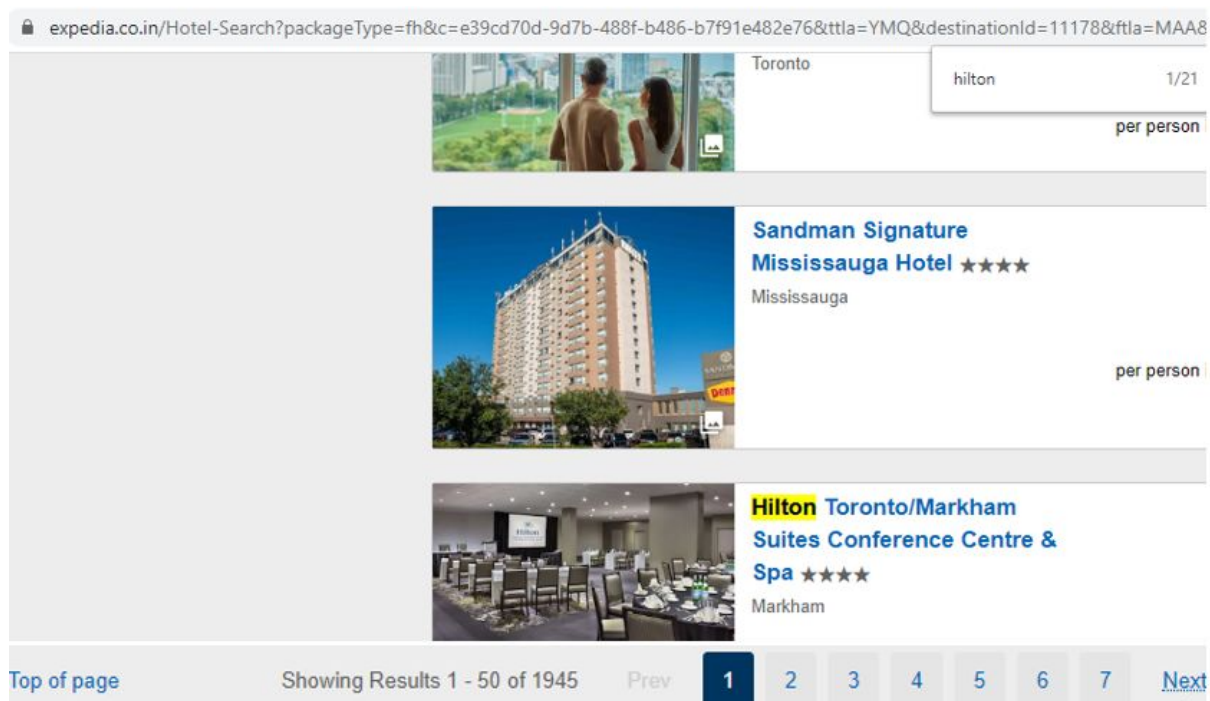


Figure 6

However we do have the keyword 'Hilton' on search results page 1, see below. The question arises is, why did selenium did not click the below hotel? The answer is, we had disabled the implicit wait timeout. So what happens is that, the script does NOT wait for 25 seconds implicit wait global timeout. So, as soon as the script finds the keyword 'Hilton', it clicks that hotel (in our case, hotel seen in figure#5)





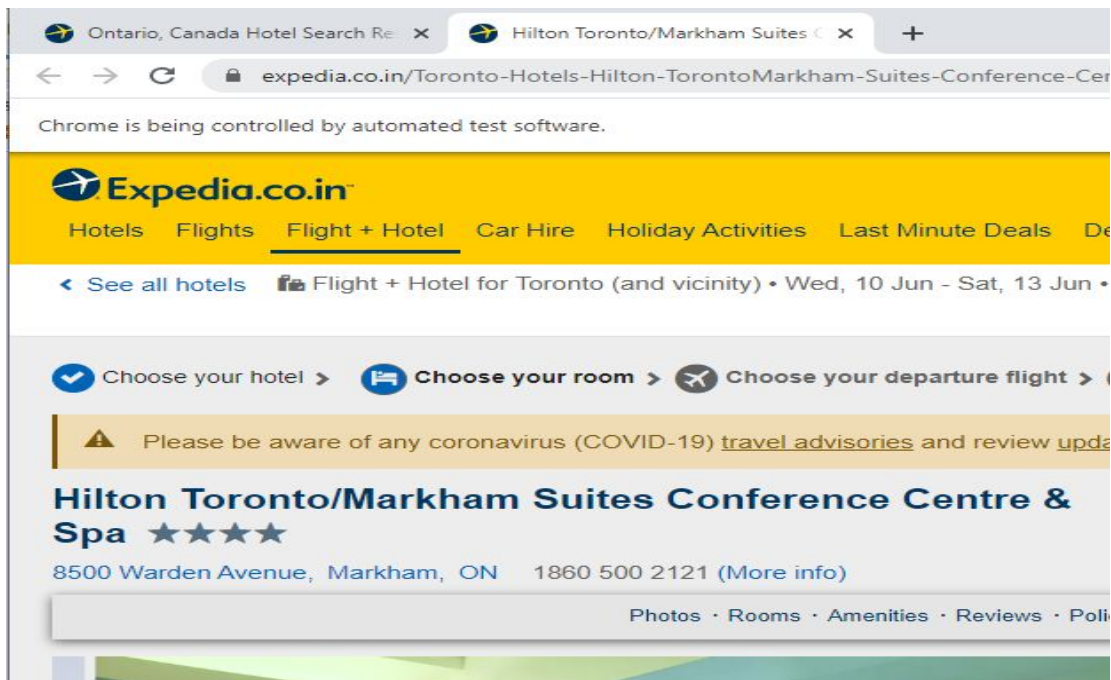
**Figure 7**

Let us enable implicit wait timeout by uncommenting it, see below

```
ExplicitWait.java
1 package test;
2
3 import java.util.concurrent.TimeUnit;
4 import org.openqa.selenium.By;
5 import org.openqa.selenium.Keys;
6 import org.openqa.selenium.WebDriver;
7 import org.openqa.selenium.chrome.ChromeDriver;
8 import org.openqa.selenium.support.ui.ExpectedConditions;
9 import org.openqa.selenium.support.ui.WebDriverWait;
10
11 public class ExplicitWait {
12
13     public static void main(String[] args) throws InterruptedException {
14         System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\DELL\\De
15
16         WebDriver driver = null;
17
18         driver = new ChromeDriver();
19
20         driver.manage().timeouts().implicitlyWait(25, TimeUnit.SECONDS);
21     }
22 }
```

**Figure 8**

Let us run the same script now. Notice that the hotel on search results page#1 gets clicked now (hotel seen in figure#7). Please note that, when you run the script, the hotel search results might vary for you, however the concept remains the same



**Figure 9**

Comment below 3 lines now (lines#20, 43, 45)

```

20         //driver.manage().timeouts().implicitlyWait(25, TimeUnit.SECONDS);
43         //d.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//a[contains(@href, 'Hilton')]")));
44
45         //driver.findElement(By.xpath("//a[contains(@href, 'Hilton')]")).click();

```

**Figure 10**

### Assignment

In the above scenario, we were searching a hotel using keyword 'Hilton' and we were then clicking the link. Now, let us do an exercise in which we would always want to click the first hotel link in the search results page. So, if you see below, we are on page#1 and our goal is to click the first hotel link on this page



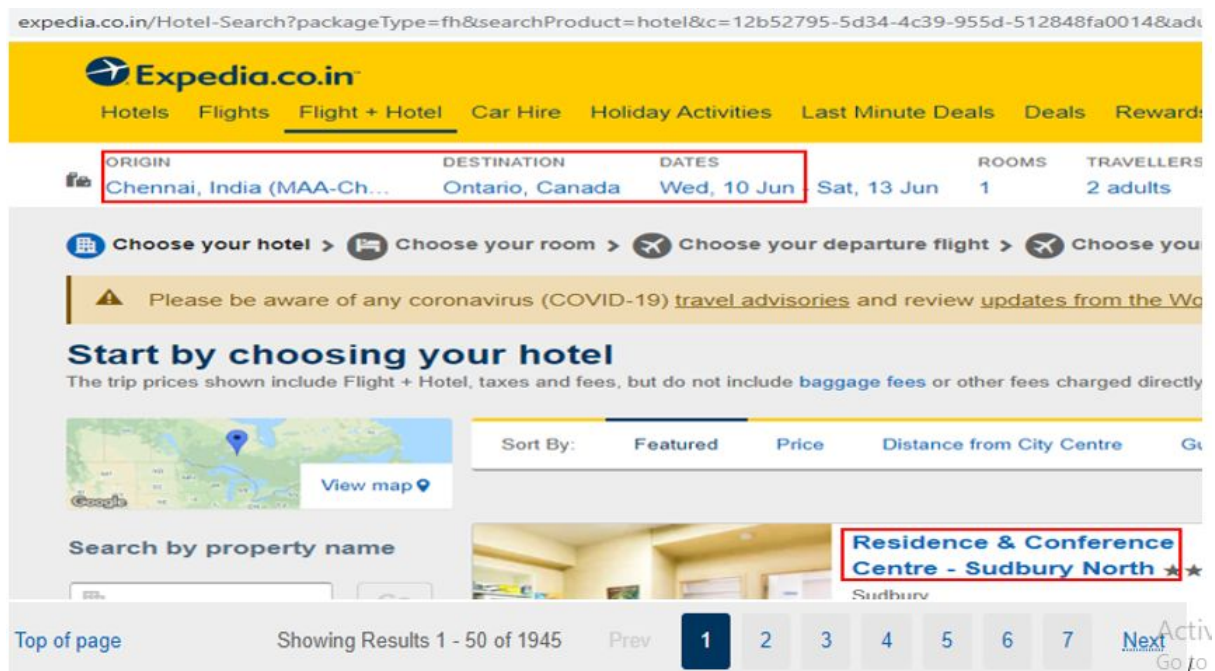


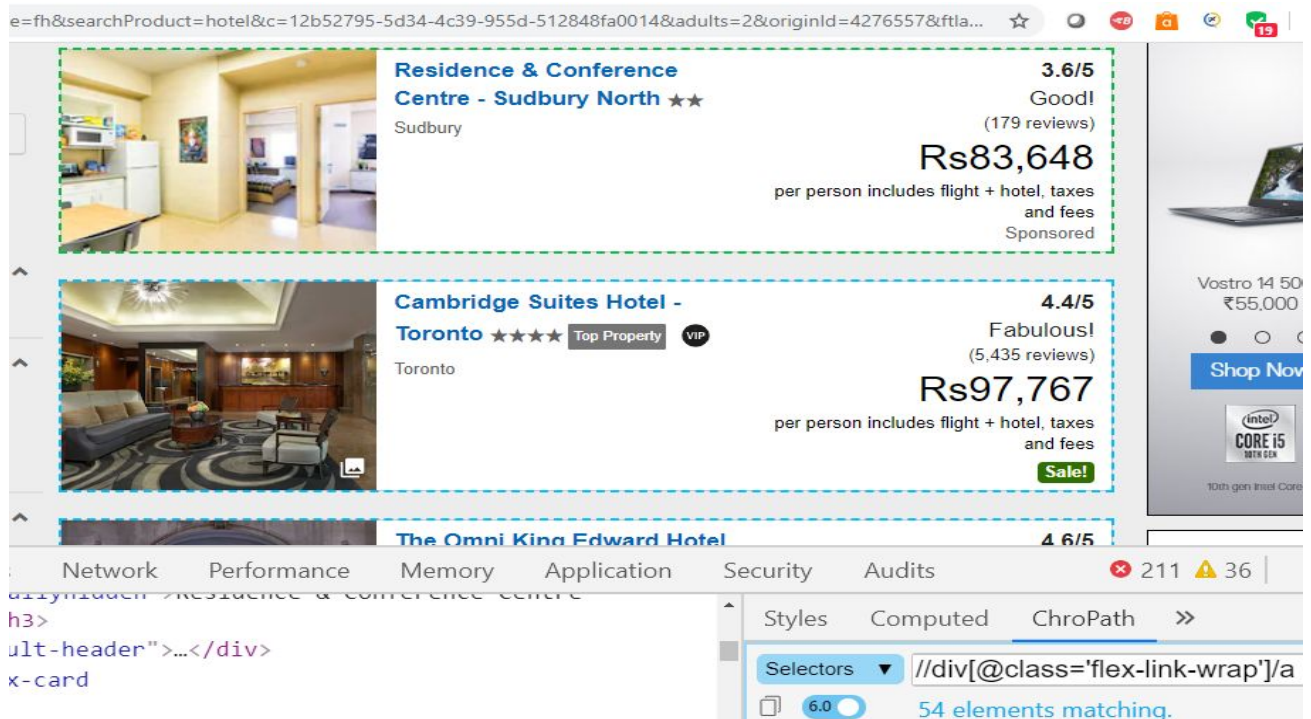
Figure 11

To do that let us find the xpath of first hotel link `//div[@class='flex-link-wrap']/a`



Figure 12

If you notice, 54 elements match to this same xpath. If we use this xpath, selenium will always click the first link (since it scans from left to right, top to bottom)



**Figure 13**

Add lines 43 & 45 that use xpath of first link. Notice that, we haven't hardcoded any hotel keyword

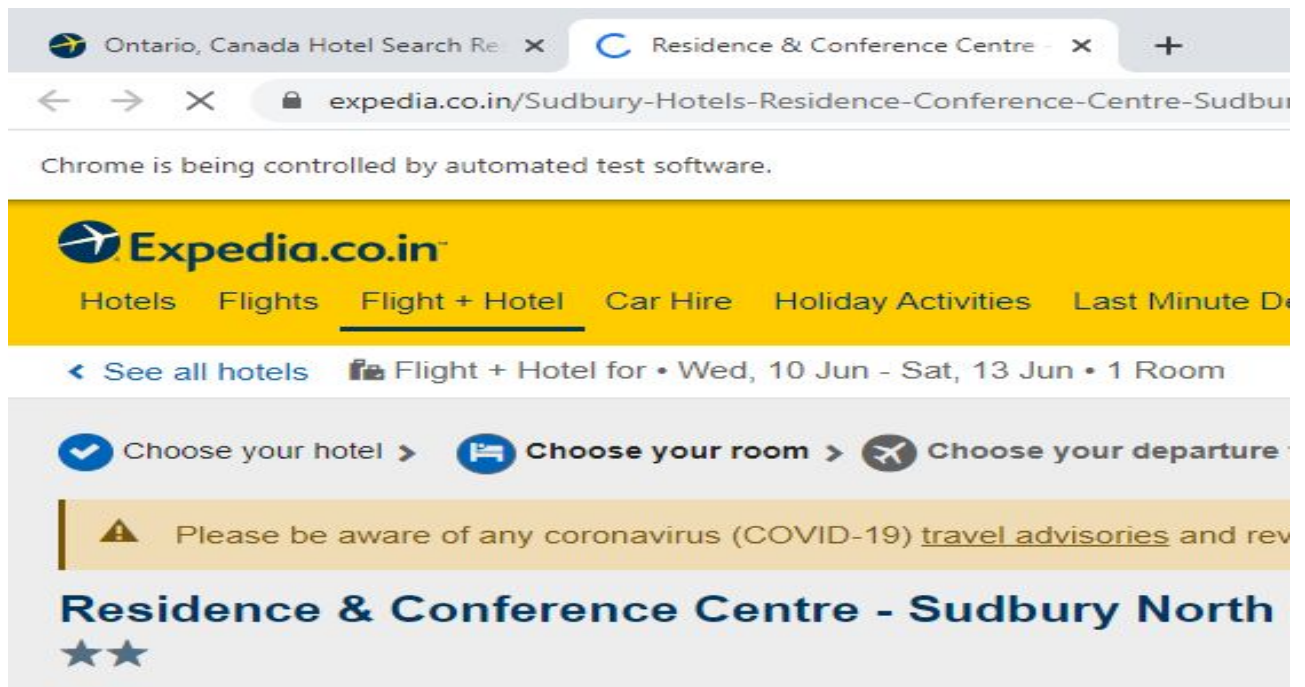
```

41
42 WebDriverWait d = new WebDriverWait(driver,25);
43 //d.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//a[contains(@href,'Hilton')]")));
44
45 //driver.findElement(By.xpath("//a[contains(@href,'Hilton')]")).click();
46
47 d.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//div[@class='flex-link-wrap']/a")));
48 driver.findElement(By.xpath("//div[@class='flex-link-wrap']/a")).click();
49

```

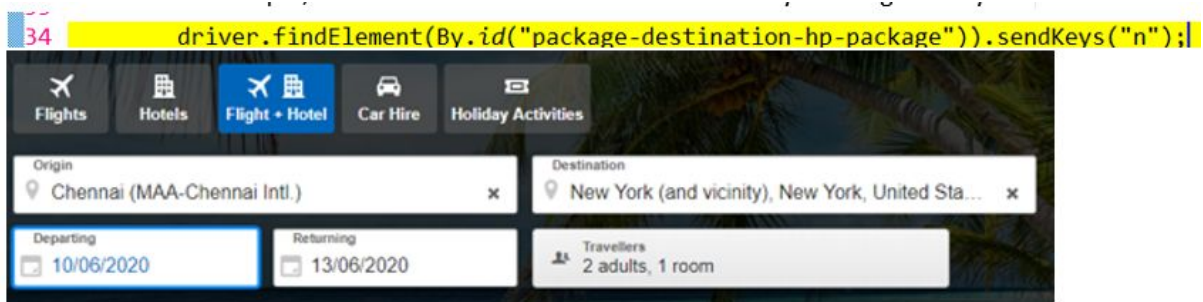
**Figure 14**

When we run the script, the search page that opens shows the first hotel link, see below



**Figure 15**

Now even if you change the origin/destination, the script will always click the first hotel link for any destination! For example, let us choose a difference destination by sending the keys 'n'



**Figure 16**

For the above search criteria, below is the first hotel that comes up (make a note of it)



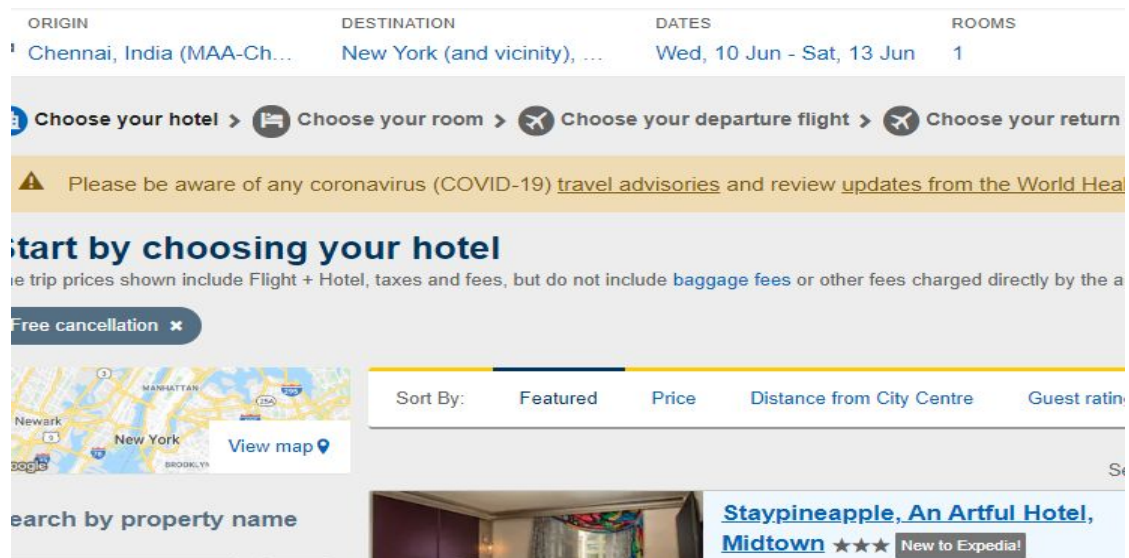


Figure 17

Run the script, notice that the same hotel link gets clicked

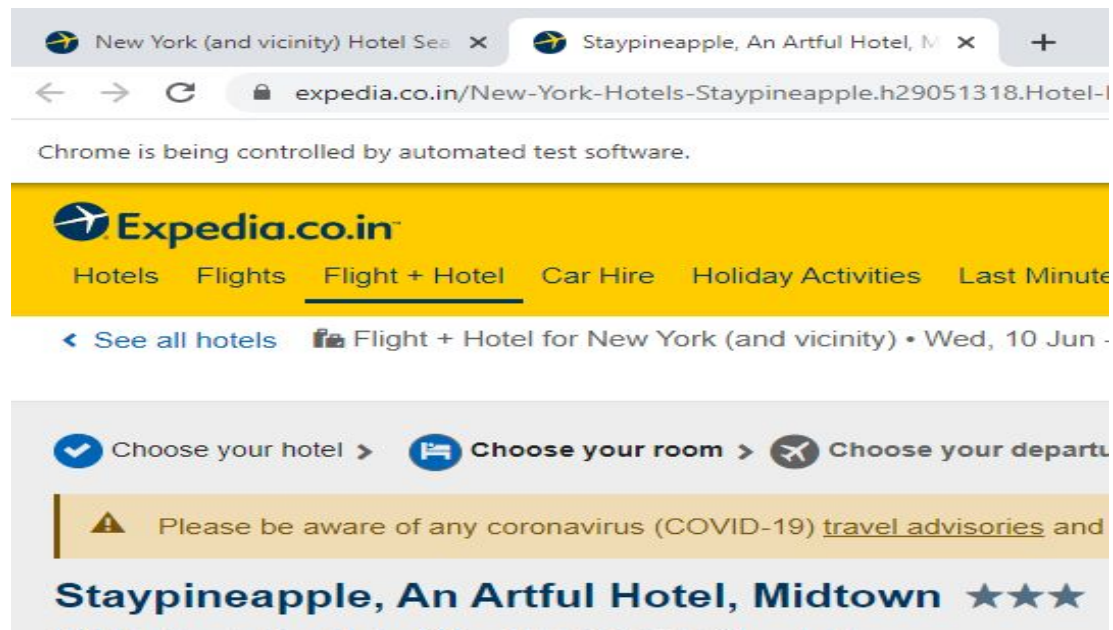


Figure 18

Let us enable implicit timeout now

```
20 driver.manage().timeouts().implicitlyWait(25, TimeUnit.SECONDS);
```

Figure 19



Run the script, same results should be seen. If you see different results, you know the reason why?

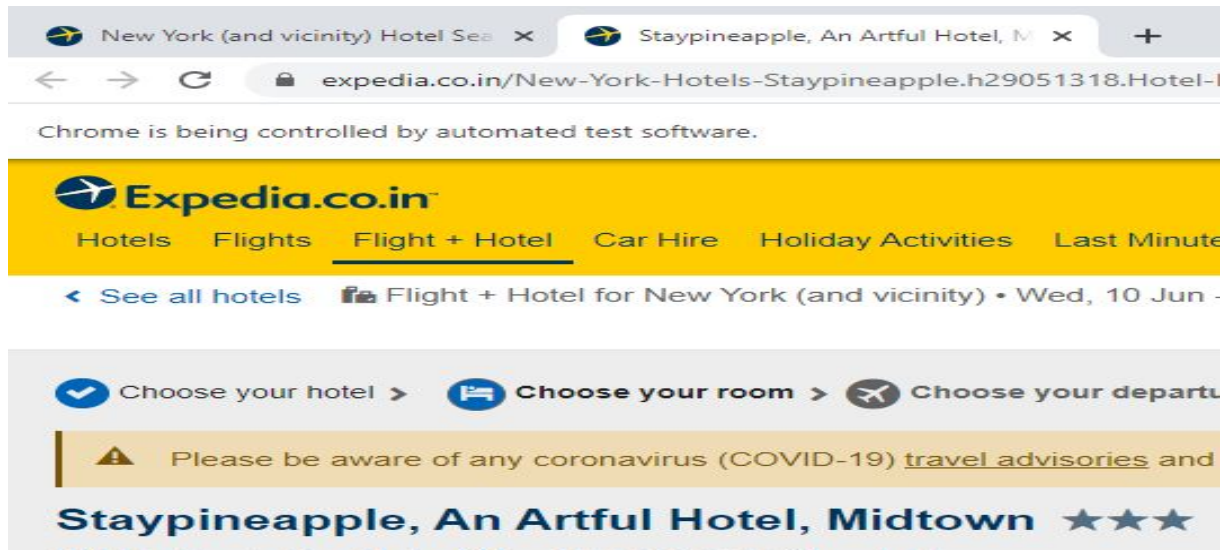


Figure 20

### Code Snippet

```
1 package test;
2
3 import java.util.concurrent.TimeUnit;
4 import org.openqa.selenium.By;
5 import org.openqa.selenium.Keys;
6 import org.openqa.selenium.WebDriver;
7 import org.openqa.selenium.chrome.ChromeDriver;
8 import org.openqa.selenium.support.ui.ExpectedConditions;
9 import org.openqa.selenium.support.ui.WebDriverWait;
10
11 public class ExplicitWait {
12
13     public static void main(String[] args) throws InterruptedException {
14         System.setProperty("webdriver.chrome.driver", "C:\\Users\\DELL\\Desktop\\TRAINING\\Software\\chromedriver.exe");
15
16         WebDriver driver = null;
17
18         driver = new ChromeDriver();
19
20         driver.manage().timeouts().implicitlyWait(25, TimeUnit.SECONDS);
21
22         driver.navigate().to("https://www.expedia.co.in/");
23
24         driver.findElement(By.xpath("//button[@data-lob='package']/span/span[2]")).click();
25
26         driver.findElement(By.id("package-origin-hp-package")).sendKeys("c");
27
28         Thread.sleep(2000);
29         driver.findElement(By.id("package-origin-hp-package")).sendKeys(Keys.ARROW_DOWN);
30         driver.findElement(By.id("package-origin-hp-package")).sendKeys(Keys.ENTER);
31         driver.findElement(By.id("package-origin-hp-package")).sendKeys(Keys.TAB);
32         driver.findElement(By.id("package-origin-hp-package")).sendKeys(Keys.TAB);
33         Thread.sleep(2000);
34
35         driver.findElement(By.id("package-destination-hp-package")).sendKeys("n");
36         Thread.sleep(2000);
37         driver.findElement(By.id("package-destination-hp-package")).sendKeys(Keys.ARROW_DOWN);
38         driver.findElement(By.id("package-destination-hp-package")).sendKeys(Keys.ENTER);
39         Thread.sleep(1000);
40         driver.findElement(By.id("package-departing-hp-package")).sendKeys("10/06/2020");
41         driver.findElement(By.id("package-departing-hp-package")).sendKeys(Keys.ENTER);
42
43         WebDriverWait d = new WebDriverWait(driver, 25);
44         //d.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//a[contains(@href, 'Hilton')]")));
45         //driver.findElement(By.xpath("//a[contains(@href, 'Hilton')]")).click();
46
47         d.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//div[@class='flex-link-wrap']/a")));
48         driver.findElement(By.xpath("//div[@class='flex-link-wrap']/a")).click();
49     }
}
```

Figure 21

So this is how we use explicit wait concept. Thank you for reading!