

Selenium - Implicit, Explicit and Fluent Wait

03 April 2017 PM 04:38

Implicit Wait

The Selenium WebDriver provides an implicit wait for synchronizing tests. When an implicit wait is implemented in tests, if WebDriver cannot find an element in the Document Object Model (DOM), it will wait for a defined amount of time for the element to appear in the DOM.

In other terms, an implicit wait polls the DOM for a certain amount of time when trying to find an element or elements if they are not immediately available. The default setting is 0.

Once set, the implicit wait is set for the life of the WebDriver object's instance.

However, an implicit wait may slow down your tests when an application responds normally, as it will wait for each element appearing in the DOM and increase the overall execution time.

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

Until the end of a test or an implicit wait is set back to 0, every time an element is searched using the findElement() method, the test will wait for 10 seconds for an element to appear.

– *com.c2t.wait.implicit.ImplicitWait*

Output

```
className=input  
Elapsed milliseconds-1: 173  
Elapsed milliseconds-2: 6040  
Elapsed seconds-2: 6.04
```

Conclusion

i.e. if it cannot find element, it will wait for 5 seconds for the element. This will happen for all the elements. If it can find element in less than 5 seconds, it will move ahead.

Explicit Wait

- The explicit wait is used to tell the Web Driver to wait for certain conditions (**Expected Conditions**) or the maximum time exceeded before throwing an **"ElementNotVisibleException"** exception.
- The explicit wait is an intelligent kind of wait, but it can be applied only for specified elements.
- Once we declare explicit wait we have to use **"ExpectedConditions"**
- An explicit wait can only be implemented in cases where synchronization is needed and the rest of the script is working fine.
- The **WebDriverWait** object will call the **ExpectedCondition** class object every 500 milliseconds until it returns successfully.

ExplicitWait.java

```
1. WebDriverWait wait = new WebDriverWait(driver, 5);  
2. wait.until(ExpectedConditions.elementToBeClickable(driver.findElement(By.name("theButton"))));  
3. driver.findElement(By.name("theButton")).click();  
4. driver.switchTo().alert().accept();  
  
1. alertIsPresent()
```

2. elementSelectionModeToBe()
3. elementToBeClickable()
4. elementToBeSelected()
5. frameToBeAvaliableAndSwitchToIt()
6. invisibilityOfTheElementLocated()
7. invisibilityOfElementWithText()
8. presenceOfAllElementsLocatedBy()
9. presenceOfElementLocated()
10. textToBePresentInElement()
11. textToBePresentInElementLocated()
12. textToBePresentInElementValue()
13. titleIs()
14. titleContains()
15. visibilityOf()
16. visibilityOfAllElements()
17. visibilityOfAllElementsLocatedBy()
18. visibilityOfElementLocated()

► Difference between Implicit Wait Vs Explicit Wait

Implicit Wait	Explicit Wait
• Implicit Wait time is applied to all the elements in the script	• Explicit Wait time is applied only to those elements which are intended by us
• In Implicit Wait, we need not specify "ExpectedConditions" on the element to be located	• In Explicit Wait, we need to specify "ExpectedConditions" on the element to be located
• It is recommended to use when the elements are located with the time frame specified in implicit wait	• It is recommended to use when the elements are taking long time to load and also for verifying the property of the element like(visibilityOfElementLocated, elementToBeClickable,elementToBeSelected)

=====

Fluent Wait

=====

<http://www.techbeamers.com/webdriver-fluent-wait-command-examples/>