iamneo

Selenium Webdriver(Synchronization,Exceptions)

Synchronization

- Synchronize between script execution and application, we need to wait after performing appropriate actions.

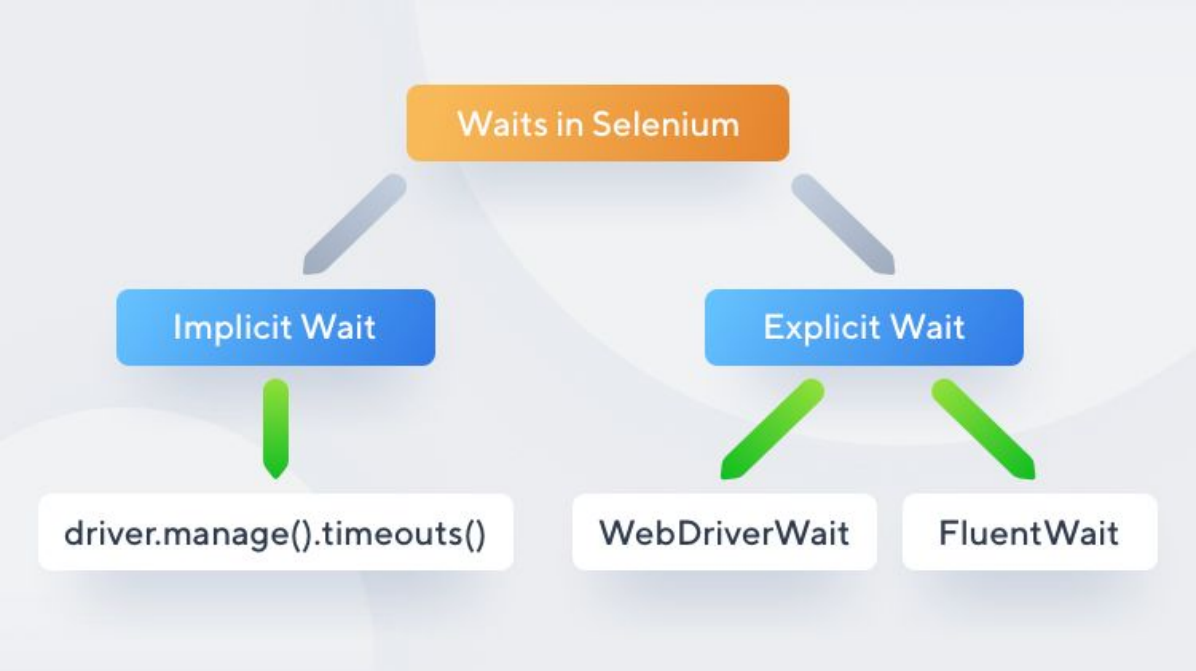
Some of Waits used in Selenium are :

1)Implicit Wait

2)Explicit Wait

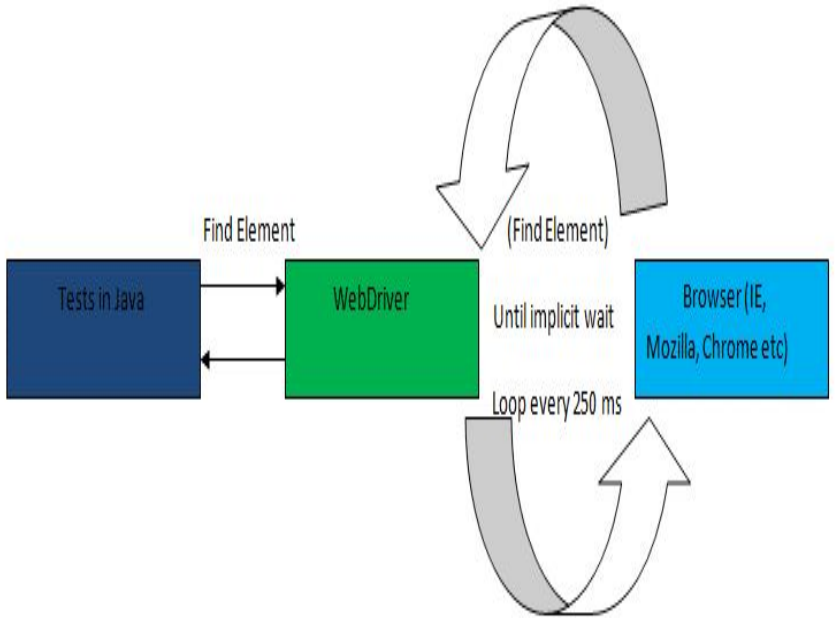
3)Fluent Wait

4)Custom Wait



Implicit wait

- ✔ An implicit wait is to tell WebDriver to poll the DOM for a certain amount of time when trying to find an element or elements if they are not immediately available.
- ✔ Once we set the time, the web driver will wait for the elemer for that time before throwing an exception.



Example

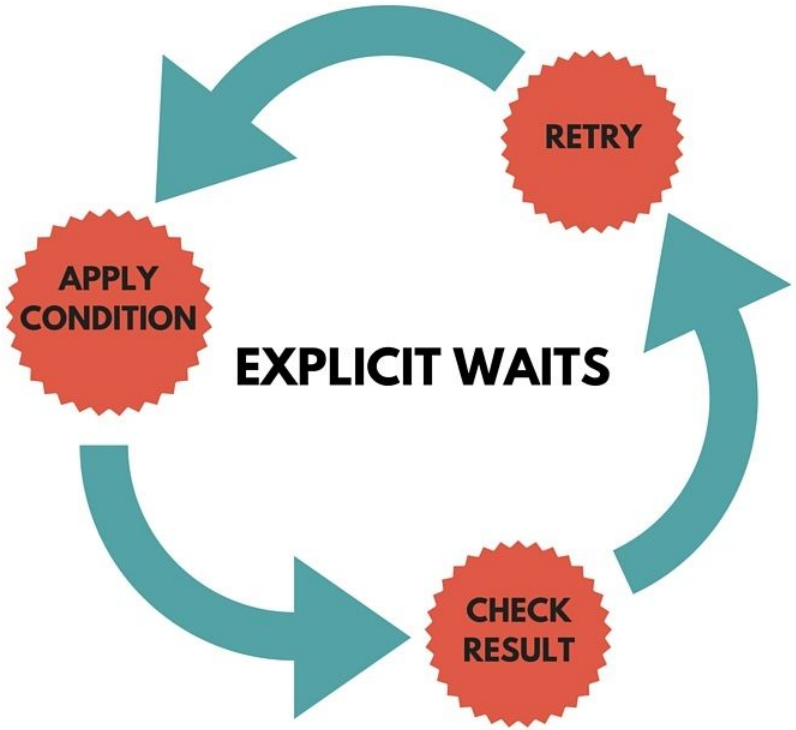
```
driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECOND);
```

Explicit wait

- ✔ The Explicit Wait in Selenium is used to tell the Web Driver to wait for certain conditions or maximum time exceeded befo throwing "ElementNotVisibleException" exception
- ✔ It can be applied only for specified elements

Example

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



Syntax for Implicit and Explicit wait

- ✔ Implicit Wait Syntax :

  “driver.manage().timeouts().implicitlyWait(TimeOut, TimeUnit.SECONDS)”

- ✔ Explicit Wait Syntax :

  1)WebElement element;

  2)WebDriverWait wait = new WebDriverWait(WebDriverReference, TimeOut);

  3)element=wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("Locator")));

  4)element.click();

Exception Handling in Selenium Webdriver

An exception is an error that happens at the time of execution of a program. However, while running a program, programming languages generates an exception that should be handled to avoid your program to crash.

The exception indicates that, although the event can occur, this type of event happens infrequently. When the method is not able to handle the Exception, it is thrown to its caller function.

Some important standard using which you can handle Exceptions in Selenium WebDriver:

- Try-catch
- Multiple catch blocks

Note:Already learnt in Java .