Waits in Selenium

- Most of the web elements are written using Java script or AJAX
- They may take certain amount of time to load on page.
- In this situation chances of getting ‘NoSuchElementException’ are more.
- To avoid such exceptions, Selenium has provided two types of Waits viz **Implicit Wait** and **Explicit Wait.**

1. Implicit Wait:

- This wait is set for the lifetime of WebDriver instance.
- Wherever we use WebDriver instance before that line wait will be applied.
- This wait, polls for 500milliseconds (*This time depends on the browsers implementation of webdriver. For firefox it is 500milliseconds*).
- If element becomes available in first 500milliseconds then element will be returned else it will wait for next 500 milliseconds. Our script will wait for max timeout.
- We can specify default time out.
- If element is not available in timeout value then ‘NoSuchElementException’ will be thrown.
- We can specify timeouts in terms of: NanoSeconds, MicroSeconds, MilliSeconds, Seconds, Minutes, Hours and Day.
- **E.g**

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

- In above example timeout value is 10 Seconds.
- Now our script will wait for min 500 milliseconds and maximum 10 Second before each and every WebElement that we wish to find using ‘*driver*’ instance.
- Eg.

```
public class ImplicitWaitDemo {
	public static void main(String[] args) {
		
		WebDriver driver;
		driver=new FirefoxDriver();
		driver.manage().timeouts().implicitlyWait(10, 
TimeUnit.SECONDS);
		driver.findElement(By.xpath("Element_1")).click();
		driver.findElement(By.xpath("Element_2")).click();
		driver.findElement(By.xpath("Element_3")).click();
		
	}
}
```

- In above example, our script will wait before clicking all three elements (*Element_1, Element_2 and Element_3*).
- Minimum wait time is 500milliseconds and maximum wait time is 10 seconds

2. Explicit Wait:

- This wait is more customized than implicit wait.
- We can apply this wait for particular web element.
- We can even customize the polling time in explicit wait.
- We can also ignore the exceptions that we don’t want while locating an element.
- Same as implicit wait, we can set timeout for the web element.
- We can also make our script wait until certain condition occurs.

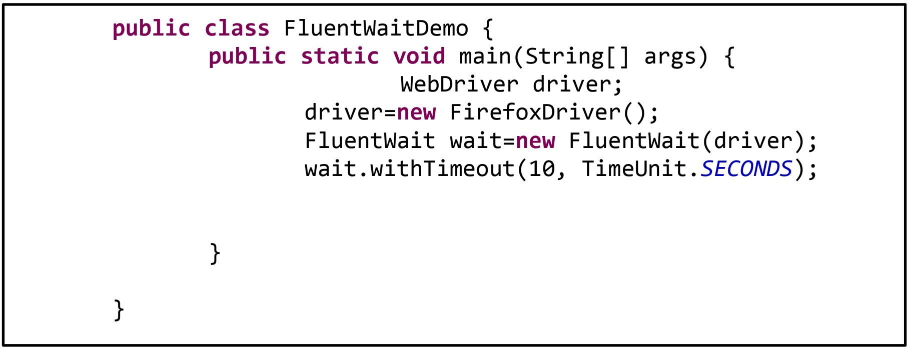
- We can achieve explicit wait using two classes of Selenium: WebDriverWait and FluentWait.

1. **<u>Using FluentWait:</u>**

- FluentWait class implements *Wait* interface.
- We can set our own timeout and polling time using FluentWait.
- We can also ignore exceptions while waiting for particular WebElement.
- We can wait till certain condition of a web element.
- This wait can be applied on particular web element unlike ‘Implicit Wait’ which applied for entire life span of WebDriver instance.

Methods of FluentWait class:

1. withTimeout(long duration, TimeUnit unit)
    - ○ This method is used to set maximum time that script will wait.
    - ○ This method takes two parameters: long duration and TimeUnit in terms of milliseconds, microseconds, seconds, hours, days etc.
    - ○ Eg.



```
public class FluentWaitDemo {
        public static void main(String[] args) {
                        WebDriver driver;
                driver=new FirefoxDriver();
                FluentWait wait=new FluentWait(driver);
                wait.withTimeout(10, TimeUnit.SECONDS);


        }

}
```

- In above example, wait is configured for maximum 10 seconds. If element is not available in 10 seconds, then NoSuchElmentException will be thrown.

2. pollingEvery(long duration, TimeUnit time):
    - This method is used to set polling time.
    - Polling time is the frequency with which our script will check for web element.
    - This method also takes two arguments: duration and time
    - Eg.

```
public class FluentWaitDemo {
       public static void main(String[] args) {
                     WebDriver driver;
              driver=new FirefoxDriver();
              FluentWait wait=new FluentWait(driver);
              wait.pollingEvery(2, TimeUnit.SECONDS);

       }

}
```

- In above example, wait is configured for 2 seconds as polling time.
- This script will search web element after every 2 seconds till maximum timeout.

3. ignoring(exceptionType):
    - This method will ignore specific types of exceptions while waiting for a condition.
    - Eg.

```
wait.ignoring(NoSuchElementException.class);
```

- Above script will ignore NoSuchElementException while searching a web element.

4. until(ExpectedConditions):

    - After configuring wait using until() method, out script will keep on waiting until one of the following encounters:
        - Until Expected condition is encountered.
        - Until the timeout expires
        - Until the current thread is interrupted
    - Eg.

```
wait.until(ExpectedConditions.alertIsPresent());
```

- Above script will wait till alert is present on web page.

2. **Using WebDriverWait:**

    - It is a specialized version of FuentWait.
    - All its constructors take WebDriver instance as parameter.
    - It inherits almost all methods of FluentWait class.
    - So it is recommended to use FluentWait class when explicit wait is required.
    - But when we are dealing with WebDriver instance, it is recommended to use WebDriverWait class.
    - It has three constructors.
        1. WebDriverWait(WebDriver driver, long timeOutInSeconds)
        2. WebDriverWait (WebDriver driver, long timeOutInSeconds, long sleepInMillis)
        3. WebDriverWait(WebDriver driver,Clock clock,Sleeper sleeper,long timeOutInSeconds, long sleepTimeOut)

Interview Questions

1. What are different waits in Selenium?

2. What is the difference between Implicit Wait and Explicit Wait ?

3. What kind of wait you have used in your projects?

4. Can you write the syntax of Fluent Wait?

5. Can you write syntax of Explicit Wait ?