
Explicit Wait

Explicit waits are used to halt the execution till the time a particular condition is met or the maximum time has elapsed.

- Unlike Implicit waits, explicit waits are applied for a particular instance only.
- WebDriver introduces classes like `WebDriverWait` and `Expected Conditions` to enforce explicit waits into the test scripts.

Explicit waits are implemented using `WebDriverWait` and `Expected Conditions` class.

To implement the explicit wait,

First we need to create the object of `WebDriverWait` class, which takes two arguments.

1. Object of `WebDriver` reference variable.
2. Time out

Syntax:

```
4 driver = webdriver.Chrome()  
5  
6 WebDriverWait(driver, 30)  
7
```

- `WebDriverWait` provides two methods
 1. `Until`
 2. `Until_not`

Until -> Calls the method provided with the driver as an argument until the return value is not false.

Until_not -> Calls the method provided with the driver as an argument until the return value is not false.

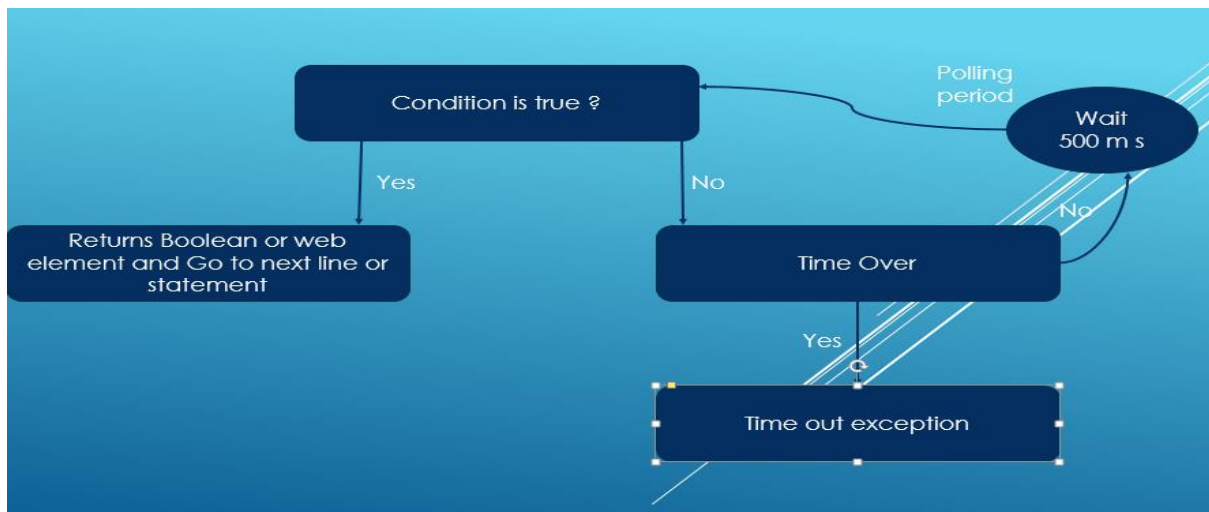
- `Until` method takes one argument called condition methods.
- To pass the condition to the **until** method we use **Expected Condition class**.

Expected Condition

There are some common conditions that are frequently came across when automating web browsers. Listed below are implementations of each. All the below mentioned condition methods are static method.

The conditions are present in a class called expected conditions. These conditions are also called as predicate.

1. Title is
2. Title contains
3. Alert is present
4. Visibility of element located
5. Element to be clickable
6. Element to be selected
7. Invisibility of element located
8. Presence of element located
9. Text to be present in element
10. Text to be present in element value
11. URL changes
12. URL contains
13. Frame to be available switch to it
14. New window is opened
15. Number of windows to be
16. Presence of all elements located ETC.



Explanation:

1. When the control comes to wait until statement it will check the condition.
2. If the condition is true it will go to next statement.
3. If the condition is false it will check for the timeout, if the time is over we get the timeout exception.
4. If the time is not over it will wait for 500 mille seconds and it will continue to check the condition.
5. Even after the specified time, if the condition is not satisfied it will throw the time out exception.

Note: When the condition is fails, explicit wait will always throw the time out exception for all the conditions.

When you are using explicit wait following are the two packages which we need to import.

```
from selenium import webdriver
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
```

Can we handle no such element exception using explicit wait?

Or

Can we handle synchronization of find element method using explicit wait?

Answer: Yes

- **title_is**

- This condition takes string as a parameter.
- The wait applies the condition to the parameter.
- If the result of applying the condition is true, **true** is returned as result.
- If the result is **false**, the wait tries the condition again after a short delay.
- While the explicit wait applies the expected condition, the condition code may generate **time out exceptions**.

```
from selenium import webdriver
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.support.wait import WebDriverWait

driver = webdriver.Chrome()
driver.get('https://demo.actitime.com/')

driver.find_element_by_id("username").send_keys("admin")
driver.find_element_by_name("pwd").send_keys("manager")
driver.find_element_by_id("LoginButton").click()

wait = WebDriverWait(driver, 30)
status = wait.until(EC.title_is("actiTIME - Enter Time-Track"), "Verifying the title")
print(status)

driver.quit()
```

- **title_contains**

- expect to check that the title contains a sub-string (parameter: title)
- Returns **"True"**, if the page contains the title or else throw Timeout exception.

```
from selenium import webdriver
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.support.wait import WebDriverWait

driver = webdriver.Chrome()
driver.get('https://demo.actitime.com/')

driver.find_element_by_id("username").send_keys("admin")
driver.find_element_by_name("pwd").send_keys("manager")
driver.find_element_by_id("LoginButton").click()

wait = WebDriverWait(driver, 30)
status = wait.until(EC.title_contains("actiTIME - Enter Time-Track"), "Verifying the title")
print(status)

driver.quit()
```

- **element_to_be_clickable**

- expect for the element to become visible and enabled so that you can click it (parameter: (By, "value"))
- Returns "**Web Element**", if the element is visible, enabled and clickable else throws Timeout exception.

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.support.wait import WebDriverWait

driver = webdriver.Chrome()
driver.maximize_window()
driver.implicitly_wait(30)
driver.get('https://demo.actitime.com/')

driver.find_element_by_id("username").send_keys("admin")
driver.find_element_by_name("pwd").send_keys("manager")
driver.find_element_by_id("LoginButton").click()

wait = WebDriverWait(driver, 30)
logoutLink = wait.until(EC.element_to_be_clickable((By.ID, "LogoutLink")), "Checking is element is clickable")
logoutLink.click()

driver.quit()
```

- **presence_of_element_located**

- expect to check that an element is present on the DOM of a page
- (parameter: locator)
- Returns "**Web Element**", if the element is present on the web page else throws Timeout exception.

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.support.wait import WebDriverWait

driver = webdriver.Chrome()
driver.maximize_window()
driver.implicitly_wait(30)
driver.get('https://demo.actitime.com/')

driver.find_element_by_id("username").send_keys("admin")
driver.find_element_by_name("pwd").send_keys("manager")
driver.find_element_by_id("LoginButton").click()

wait = WebDriverWait(driver, 30)
logoutLink = wait.until(EC.presence_of_element_located((By.ID, "LogoutLink")), "Checking is element is present")
logoutLink.click()

driver.quit()
```

Difference between implicit wait and explicit wait

Implicit wait	Explicit wait
Waiting condition is built-in	We should mention the waiting condition
We can handle synchronization of all find element and find elements	We can handle synchronization of any methods but one at a time
If time is over we get no such element exception	If time is over we get time out exception