Implicit & Explicit Waits in selenium python

Table of content

Waits in Selenium Python

These days most web apps are using AJAX techniques and angular. When the browser loads a page, the elements within that page may load at different time intervals. This makes locating elements difficult: if an element is not yet present in the DOM, a find element function will raise an ElementNotVisibleException exception.

Using waits, we can solve this issue. Waiting provides some slack between locating an element and operating on the element.

Selenium Python provides two types of waits - **implicit & explicit**. An explicit wait makes selenium wait for a specific condition to occur before proceeding further with execution.

An implicit wait makes selenium python poll the DOM for a certain amount of time with a 300ms interval when trying to locate an element.

Implicit Wait in Python Selenium

Selenium Python tries to find the element without bothering about whether elements are loaded or not, and selenium python throws NoSuchElementException if the element is not present.

Implicitly wait is one of the ways to request selenium not throw any exception until provided time. **A default wait time of the selenium is 500 milliseconds**, implicitly wait overrides the default wait time of the selenium python.

If the element is found before implicitly wait time, selenium moves to the next commands in the program without waiting further, this wait is also called dynamic wait.

The implicit wait is set for the entire duration of your selenium driver and is set at the start of your program. Most of the automation tester writes the implicit wait after the creation of the browser object.

Let's consider the implicit wait of 30 seconds, Implicit wait tries to find the element in the first go, if the element is not present implicit wait tries to find the element after 300ms of first polling if the element is not available on the second time also, then implicit wait tries the third time after 300 ms of the second try and it goes on till the time reaches the 30 seconds.

What it does is, if your selenium python doesn't find any element then instead of throwing an exception, the implicit wait makes your driver wait for the

specified wait time and then try to find the element once again till the time limit is reached.

If the driver still does not find the element, then it throws an exception. Implicit wait does the same for all the elements in your program, so you just have to set it once.

```
driver = webdriver.Chrome(executable_path=r'D:PATHchromedriver.exe');
driver.implicitly_wait(30)
```

Explicit Wait in selenium Python

The explicit wait is used to tell the Web Driver to wait for specific conditions or the maximum time limit before throwing an Exception.

We can reuse the WebdriverWait object once we create it. The explicit wait will be applicable for only one line, and we have to use it with ExpectedConditions class.

ExplicitWait does not have any effect on find element and find elements. ExplicitWait also called WebdriverWait. WebDriverWait by default calls the ExpectedCondition every 500 milliseconds until it returns successfully.

```
Syntax : wait=new WebDriverWait( driver, timeoutInSeconds);
```

Explicit waits are an excellent way to organize a test script and provide more flexibility, by allowing us to design out tests in a way, where we can wait for some predefined or custom conditions and then carry on with what we want.

Below code waits for the element to become clickable

```
driver = webdriver.Chrome(executable_path=r'D:PATHchromedriver.exe');
driver.get("https://google.com");
wait = new WebDriverWait(driver, 30 /*timeout in seconds*/);
wait.until(ExpectedConditions.element_to_be_clickable(By.xpath("//button[@id='btn1']")));
```

Below are few Expected Conditions :

- title_is : An expectation for checking the title of a page. The title is the expected title, which must be an exact match returns True if the title matches, false otherwise.
- title_contains : An expectation for checking that the title contains a case-sensitive substring. The title is the fragment of title expected returns True when the title matches, False otherwise
- presence_of_element_located : An expectation for checking that an element is present on the DOM of a page. This does not necessarily mean that the element is visible. Locator - used to find the element returns the WebElement once it is located
- visibility_of_element_located : An expectation for checking that all elements are present in the DOM of a page and visible. Visibility means that the elements are not only displayed but also has a height and width that is greater than 0. locator - used to find the elements returns the list of WebElements once they are located and visible
- visibility_of : An expectation for checking that an element, known to be present in the DOM of a page, is visible. Visibility means that the element is not only displayed but also has a height and width that is greater than 0. element is the WebElement returns the (same) WebElement once it is visible
- presence_of_all_elements_located : An expectation for checking that there is at least one element present on a web page. locator is used to find the element returns the list of WebElements once they are located
- text_to_be_present_in_element : An expectation for checking if the given text is present in the specified element. locator, text

- text_to_be_present_in_element_value : An expectation for checking if the given text is present in the element's locator, text
- frame_to_be_available_and_switch_to_it : An expectation for checking whether the given frame is available to switch to. If the frame is available, it switches the given driver to the specified frame
- invisibility_of_element_located : An Expectation for checking that an element is either invisible or not present in the DOM. Locator used to find the element
- element_to_be_clickable : An Expectation for checking an element is visible and enabled such that you can click it.
- staleness_of : Wait until an element is no longer attached to the DOM. Element is the element to wait for. Returns False if the element is still attached to the DOM, true otherwise.
- element_to_be_selected : An expectation for checking the selection is selected. Element is WebElement object
- element_located_to_be_selected : An expectation for the element to be located is selected. locator is a tuple of (by, path)
- element_selection_state_to_be : An expectation for checking if the given element is selected. Element is WebElement object is_selected is a Boolean
- element_located_selection_state_to_be : An expectation to locate an element and check if the selection state specified is in that state. Locator is a tuple of (by, path) is_selected is a boolean
- new_window_is_opened : An expectation that a new window will be opened and have the number of windows handles increase
- number_of_windows_to_be : An expectation for the number of windows to be a certain value.

----END----

Muntasir Abdullah Mizan
muntasir.abdullah01@gmail.com