Cognizant | Academy

Business Focused. Learner Centric.

Selenium Waits

Selenium - Waits

Explicit and Implicit Waits

Why?

- To elapse the automated task execution for a certain amount of time before continuing with the next step.

When?

- While navigating from one page to another.
- Waiting for the Page title.
- Wait till element is displayed.

Explicit and Implicit Waits

Implicit Wait :

- To tell Webdriver to poll the DOM for a certain amount of time when trying to find an element(s) 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 instance.

```
Webdriver driver = new FirefoxDriver();
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
driver.get("http://somedomain/url_that_delays_loading");
WebElement myDynamicElement =
driver.findElement(By.id("myDynamicElement"));
```

Explicit and Implicit Waits

Explicit Wait :

- Is the code you define to wait for a certain condition to occur before proceeding further in the code.
- The simplest form of this is Thread.sleep().
- Explicit wait can be implemented using WebdriverWait in combination with Expected Condition.

Expected Conditions :

- Contains a set of predefined conditions to use with WebdriverWait in Java.

Explicit and Implicit Waits

Example :

- Wait till Element is Clickable - it is Displayed and Enabled

```
WebdriverWait wait = new WebdriverWait(driver, 10);
WebElement element =
wait.until(ExpectedConditions.elementToBeClickable(By.id("someid")));
```

- This waits up to 10 seconds before throwing a TimeoutException.
- If it finds the element will return it in 0 - 10 seconds.
- WebdriverWait by default calls the ExpectedCondition every 500 milliseconds until it returns successfully

FluentWait

- Defines the maximum amount of time to wait for a specific condition and frequency.
- Using this the condition can be checked before throwing an “ElementNotVisibleException” exception.
- Used when we have web elements which sometimes is visible in few seconds and sometimes takes more time than usual.

FluentWait

- Mainly used in Ajax applications.

- We could set the default pooling period based on our requirement.

- We could ignore any exception while polling an element.

- Syntax:

```
wait wait = new FluentWait(Webdriver reference)
.withTimeout(timeout, SECONDS)
.pollingEvery(timeout, SECONDS)
.ignoring(Exception.class);
```

FluentWait

Example :

```
FluentWait<Webdriver> wait = new
FluentWait<Webdriver>(driver)
.withTimeout(45, TimeUnit.SECONDS)
.pollingEvery(5, TimeUnit.SECONDS)
.ignoring(NoSuchElementException.class);
```