Review Esperas Implícita e Explícita

**O que você precisa entender sobre as aulas de esperas Implícita e Explícita é que se você não esperar o elemento ser carregado corretamente na página HTML os nossos testes podem quebrar. Na última aula usamos duas classes, essa duas classes são responsáveis pela espera Explícita:**

1. from selenium.webdriver.support.wait import WebDriverWait
2. from selenium.webdriver.support import expected\_conditions

outra forma de importar o expected\_conditions já deixando explicito qual classe queremos usar.

1. from selenium.webdriver.support.expected\_conditions import visibility\_of\_element\_located

Somente com a classe WebDriverWait podemos fazer a espera sem usar o expected\_condition, um exemplo é esse via lambda, exemplo abaixo:

1. element = WebDriverWait(driver, 10).until(lambda x: x.find\_element\_by\_id("someId"))

Com o modulo **expected\_conditions,**existe várias classes para tratar certas condições que esperamos dos elementos, ele simplesmente trata os dados, como nesse no exemplo abaixo ele espera o elemento ser visível para o usuário realizar o teste.

1. from selenium.webdriver.common.by import By
2. from selenium.webdriver.support.wait import WebDriverWait
3. from selenium.webdriver.support import expected\_conditions

6. def espera\_explicita(driver, selector, delay=30):
7. # Espera Explicita > Melhor forma de fazer.
8. # An expectation for checking that an element is present on the DOM of a page and visible.
9. return WebDriverWait(driver, delay).until(
10. expected\_conditions.visibility\_of\_element\_located((By.CSS\_SELECTOR, selector)))

Classe no Selenium:

1. class visibility\_of\_element\_located(object):
2. """ An expectation for checking that an element is present on the DOM of a
3. page and visible. Visibility means that the element is not only displayed
4. but also has a height and width that is greater than 0.
5. locator - used to find the element
6. returns the WebElement once it is located and visible
7. """
8. def \_\_init\_\_(self, locator):
9. self.locator = locator
11. def \_\_call\_\_(self, driver):
12. try:
13. return \_element\_if\_visible(\_find\_element(driver, self.locator))
14. except StaleElementReferenceException:
15. return False

**No Módulo de expected\_conditions, temos as seguintes esperas abaixo, você pode utilizar todas essas condições:**

* **title\_is**- Verifica se o titulo da página é igual ao que foi passado na comparação, nesse primeiro caso o texto dos dois lados tem que ser exatamente igual, ele usa o "==" .
* **title\_contains**- Verifica se o titulo da página é igual ao que foi passado na comparação, nesse caso não precisa ser exatamente igual, ele usa o "in" pra comparar.
* **presence\_of\_element\_located**- Verifica a presença do elemento na DOM.
* **visibility\_of\_element\_located**- Verifica se o elemento está visível para o usuário, no caso de não encontrar o elemento retorna false.
* **visibility\_of**- Verifica se o elemento está visível para o usuário, no caso de não encontrar o elemento retorna um erro.
* **presence\_of\_all\_elements\_located**-Verifica a presença de todos os elemento na DOM.
* **text\_to\_be\_present\_in\_element**- Verifica se um determinado texto está presente na página.
* **text\_to\_be\_present\_in\_element\_value**- Verifica se um determinado texto está presente no value do elemento, geralmente em campos do tipo input.
* **frame\_to\_be\_available\_and\_switch\_to\_it**- Verifica se o frame foi carregado corretamente pra efetuar a troca.
* **invisibility\_of\_element\_located**- Verifica se o elemento não está mais visível na DOM.
* **element\_to\_be\_clickable**- Verifica se o elemento está hábil para realizar o clique.
* **staleness\_of**-  Wait until an element is no longer attached to the DOM, element is the element to wait for.
* **element\_to\_be\_selected**- Verifica se o elemento está hábil para ser selecionável.
* **element\_located\_to\_be\_selected**- Verifica se o elemento foi localizado e está hábil para ser selecionável.
* **element\_selection\_state\_to\_be**- Verificar se o elemento dado está selecionado enviando a condição desejada true/false, você deve passar o  elemento que é objeto WebElement, e a condição é um booleano, verifica se a condição é verdadeira ou falsa.
* **element\_located\_selection\_state\_to\_be**- An expectation to locate an element and check if the selection state specified is in that state.
* **alert\_is\_present *-***Verifica a presença do alerta na página.

**Referências:**

<http://selenium-python.readthedocs.io/waits.html>