Python+Selenium笔记（十）：元素等待机制

 （一） 前言

突然的资源受限或网络延迟，可能导致找不到目标元素，这时测试报告会显示测试失败。这时需要一种延时机制，来使脚本的运行速度与程序的响应速度相匹配，WebDriver为这种情况提供了隐式等待和显式等待两种机制。

（二） 隐式等待

一旦设置隐式等待时间，就会作用于这个WebDriver实例的整个生命周期（对所有的元素查找都生效），设置隐式等待时间后，Webdriver会在一定时间内持续检测和搜寻DOM，以便于查找一个或多个不是立即加载成功并可用的元素。隐式等待的默认时间是0. WebDriver使用implicitly\_wait()来设置等待时间，单位秒。超过等待时间还没找到，就报NoSuchElementException异常。

driver.implicitly\_wait(10)

（三） 显式等待

WebDriver提供了WebDriverWait类和expected\_conditions模块来实现显式等待。相比隐式等待，显示等待更加智能。显示等待就是设置一个前置条件，在等待时间内，每隔一段时间检查一次前置条件是否满足，满足则执行下一步，超时则报TimeoutException异常。

（四） WebDriverWait类

until\_not（method, message=''）：直到返回值为false，其他和until相同

（五） expected\_conditions模块

    expected\_conditions模块提供了多种定义好的前置条件，需要配合WebDriverWait使用。

（六） expected\_conditions 示例

下面的代码，try: 部分，每一部分都是独立可用的（我只是验证不同前置条件的用法后就注释掉）。另外这里只对方法的使用方式（方法的功能）进行说明，不对使用场景进行说明（比如有没有必要这么做什么的）。

（七） 示例（自定义前置条件）

expected\_conditions类提供了多种定义好的前置条件（预期等待条件），没有前置条件符合时，也可以通过WebDriverWait自定义前置条件。

下面这个是WebDriverWait类自带的部分注释。

示例：（等待博客园个人主页（点击首页的园子跳转到的页面）的下拉菜单有5个可选项）

#lambda表达式其实就是一个匿名函数，冒号左边的可以理解为函数名及参数，右边的可以理解为函数的返回值，具体可以百度python lambda

（八） 总结

应用元素等待机制，对于构建高度稳定可靠的测试是必不可少的。在使用过程中，应该尽量避免隐式等待和显示等待混合使用。至于隐式等待和显示等待的优缺点，看书上和网上一般是比较推荐使用显示等待，不过我自己试了下，暂时是没看出在运行速度方面有多大区别（可能等以后有比较丰富的项目经验后，再回头来说说隐式等待和显示等待的优缺点）。

#设置超时时间为10秒

WebDriverWait（driver, timeout, poll\_frequency=POLL\_FREQUENCY, ignored\_exceptions=None）

driver：浏览器驱动实例

timeout：等待时间，单位秒

poll\_frequency：每隔多长时间检查一次，默认0.5秒

ignored\_exceptions：忽略的异常，默认只有NoSuchElementException

until方法 和 until\_not方法：

until（method, message=''）：method指要执行的方法（等待时间内每隔一段时间，执行一次），直到返回值为true，超时则报TimeoutException异常，message将传入异常（message参数可不填）

element\_to\_be\_clickable(locator)

例如：WebDriverWait(driver,10).until(expected\_conditions.element\_to\_be\_clickable((By.NAME,'11')))

下面的都是以这种方式，只是前置条件不同，传的参数也可能不同

WebDriverWait(driver,10).until（）

等待查找的元素可见并且可用，以便可以点击，返回定位到的元素

element\_to\_be\_selected(locator)

等待直到元素被选中

invisibility\_of\_element\_located(locator)

等待一个元素在DOM中不可见 或不存在

presence\_of\_all\_elements\_located(locator)

等待至少有一个定位器查找的元素出现在网页中，返回一组元素

presence\_of\_element\_located(locator)

等待定位器查找的元素出现在网页中，或者可以在DOM中找到，返回一个被定位到的元素

text\_to\_be\_present\_in\_element(locator,text)

参数：text，指定的文本

等待元素能被定位，并且带有指定的文本信息

title\_contains(title)

参数：title，指要校验标题包含的字符串

等待网页标题包含指定的字符串，成功时返回True，否则返回false

title\_is(title)

参数：title，指要校验的标题

等待网页标题与预期一致，成功时返回True，否则返回false

visibility\_of(element)

参数：element，指一个元素

等待元素出现在DOM中，是可见的，并且宽和高都大于0，变为可见的，将返回一个元素（同一个）

visibility\_of\_element\_located(locator)

等待元素出现在DOM中，是可见的，并且宽和高都大于0，变为可见的，将返回一个元素

alert\_is\_present()

判断是否存在警告窗口

1 from selenium import webdriver  
 2 from selenium.webdriver.support.ui import WebDriverWait  
 3 from selenium.webdriver.support import expected\_conditions  
 4 from selenium.webdriver.common.by import By  
 5   
 6 driver = webdriver.Firefox()  
 7 driver.maximize\_window()  
 8 driver.get('https://www.cnblogs.com/')  
 9   
10 # try:  
11 # #等待博客园首页的【找找看】按钮可见并可用  
12 # search\_btn = WebDriverWait(driver,10).until(expected\_conditions.element\_to\_be\_clickable((By.CLASS\_NAME,'search\_btn')))  
13 # print(search\_btn.get\_attribute('value'))  
14   
15 # try:  
16 # login\_area = driver.find\_element\_by\_css\_selector('#login\_area')  
17 # login = login\_area.find\_element\_by\_link\_text('登录')  
18 # login.click()  
19 # remember\_me = driver.find\_element\_by\_id('remember\_me')  
20 # remember\_me.click()  
21 # #等待直到登录页面的复选框被选中  
22 # WebDriverWait(driver, 10).until(expected\_conditions.element\_located\_to\_be\_selected((By.ID, 'remember\_me')))  
23   
24 # try:  
25 # search\_file = driver.find\_element\_by\_id('zzk\_q')  
26 # search\_btn = driver.find\_element\_by\_class\_name('search\_btn')  
27 # search\_file.send\_keys('python')  
28 # search\_btn.click()  
29 # #网页标题是否包含 python  
30 # WebDriverWait(driver, 10).until(expected\_conditions.title\_contains('python'))  
31   
32 try:  
33 search\_file = driver.find\_element\_by\_id('zzk\_q')  
34 #检查元素是否出现在DOM中，是可见的，并且宽和高都大于0  
35 search\_file = WebDriverWait(driver,10).until(expected\_conditions.visibility\_of(search\_file))  
36 print(search\_file)  
37 finally:  
38 driver.quit()

class WebDriverWait(object):  
 def \_\_init\_\_(self, driver, timeout, poll\_frequency=POLL\_FREQUENCY, ignored\_exceptions=None):  
 """  
 Example:  
 from selenium.webdriver.support.ui import WebDriverWait \n  
 element = WebDriverWait(driver, 10).until(lambda x: x.find\_element\_by\_id("someId")) \n  
 is\_disappeared = WebDriverWait(driver, 30, 1, (ElementNotVisibleException)).\ \n  
 until\_not(lambda x: x.find\_element\_by\_id("someId").is\_displayed())  
 """

from selenium import webdriver  
from selenium.webdriver.support.ui import WebDriverWait  
from selenium.webdriver.support.ui import Select  
  
profile = webdriver.FirefoxProfile\  
 (r'C:\Users\quanhua\AppData\Roaming\Mozilla\Firefox\Profiles\tnwjkr4m.selenium')  
driver = webdriver.Firefox(profile)  
driver.maximize\_window()  
driver.get('https://home.cnblogs.com/')  
try:  
 #等待 博客园个人主页中的下拉菜单有5个可选项  
 WebDriverWait(driver,10).until(lambda l:len(Select(l.find\_element\_by\_id('sel\_application')).options) == 5)  
finally:  
 driver.quit()