Python+Selenium笔记（十二）：数据驱动测试

(一)   前言

通过使用数据驱动测试，实现对输入值和预期结果的参数化。（例如：输入数据和预期结果可以直接读取Excel文档的数据）

(二)   ddt

使用ddt执行数据驱动测试，ddt库可以将测试中的变量参数化。使用ddt的时候，在测试类上使用@ddt装饰符，在测试方法上使用@data装饰符。@data装饰符将参数当作测试数据，参数可以是单个值、列表、元组、字典。对于列表和元组，需要用@unpack装饰符把列表和元组解析成多个参数。

使用下面的命令安装ddt库

pip install ddt

(三)   通过Excel获取数据

读取Excel文件，需要用到xlrd库。

安装xlrd的库

pip install xlrd

如果要往Excel表格写数据，需要用到xlwt库

pip install xlwt

(四)   示例中用到的excel文件

(五)   示例

1 from selenium import webdriver  
 2 from ddt import ddt,data,unpack  
 3 import xlrd  
 4 import unittest  
 5 #读取excel文件的函数  
 6 def get\_data(file\_name):  
 7 rows = []  
 8 #读取excel的数据  
 9 book = xlrd.open\_workbook(file\_name)  
10 #通过索引访问第一个sheet页  
11 sheet = book.sheet\_by\_index(0)  
12 #迭代读取excel第一个sheet页的数据，sheet.nrows指excel的行数  
13 for r\_idx in range(1, sheet.nrows):  
14 #row\_values读取第r\_idx行的数据（0代表读取第1列及后面所有列的数据）  
15 #读取数据的时候，我们一般说的第一行、第一列，索引都是0  
16 #所以r\_idx=1的时候，读取的其实是excel第二行的数据  
17 rows.append(list(sheet.row\_values(r\_idx,0)))  
18 #先将手机号删除并赋值给pthone,然后转换为字符串并添加回原来的位置  
19 pthone = rows[r\_idx - 1].pop(1)  
20 rows[r\_idx - 1].insert(1, str(int(pthone)))  
21 return rows  
22 @ddt  
23 class RegisterNewUserDDT(unittest.TestCase):  
24 @classmethod  
25 def setUpClass(cls):  
26 cls.driver = webdriver.Chrome()  
27 cls.driver.implicitly\_wait(20)  
28 cls.driver.maximize\_window()  
29 cls.driver.get('https://www.cnblogs.com/')  
30 login\_area = cls.driver.find\_element\_by\_css\_selector('#login\_area')  
31 register = login\_area.find\_element\_by\_link\_text('注册')  
32 register.click()  
33 #读取excel文件的数据作为参数  
34 @data(\*get\_data('data/reTest.xlsx'))  
35 @unpack  
36 def test\_register\_new\_user(self,email,phone,login\_name,nickname,password,confirm\_password,expected\_result):  
37 driver = self.driver  
38 self.assertTrue('用户注册 - 博客园' == driver.title)  
39 # 定位注册页面各个字段  
40 user\_email = driver.find\_element\_by\_id('Email')  
41 user\_phone\_country = driver.find\_element\_by\_id('CountryCode')  
42 user\_phone = driver.find\_element\_by\_id('PhoneNum')  
43 user\_login\_name = driver.find\_element\_by\_id('LoginName')  
44 user\_nickname = driver.find\_element\_by\_id('DisplayName')  
45 user\_password = driver.find\_element\_by\_id('Password')  
46 user\_confirm\_password = driver.find\_element\_by\_id('ConfirmPassword')  
47 #清除各字段的值（如果有）  
48 user\_email.clear()  
49 user\_phone.clear()  
50 user\_login\_name.clear()  
51 user\_nickname.clear()  
52 user\_password.clear()  
53 user\_confirm\_password.clear()  
54 #输入邮箱、手机号等信息  
55 user\_email.send\_keys(email)  
56 user\_phone.send\_keys(phone)  
57 user\_login\_name.send\_keys(login\_name)  
58 user\_nickname.send\_keys(nickname)  
59 user\_password.send\_keys(password)  
60 user\_confirm\_password.send\_keys(confirm\_password)  
61 #判断提示是否正确（这边应该有办法可以获取当前是第几次执行，下面的写法太死板了）  
62 if phone == '1':  
63 phone\_error = driver.find\_element\_by\_id('PhoneNum-error')  
64 self.assertTrue(phone\_error.text == expected\_result)  
65 elif login\_name == 'b':  
66 loginName\_error = driver.find\_element\_by\_id('LoginName-error')  
67 self.assertTrue(loginName\_error.text == expected\_result)  
68   
69 @classmethod  
70 def tearDownClass(cls):  
71 cls.driver.quit()

