

Automatyzacja przypadku testowego przy pomocy Selenium Webdriver

```
tester@tester-m:~/Desktop/15-16.05$ python3 CastoramaDobiechSylwia.py  
testInvalidPassword (__main__.Registration) ... ok  
  
-----  
Ran 1 test in 70.199s  
  
OK
```

Autor:

Sylwia Dobiech

Chorzów 2021

1. Przypadek testowy:

Rejestracja na stronie sklepu Castorama

Przypadek testowy 001:

Rejestracja z wykorzystaniem błędnego hasła (brak dużych liter, znaków specjalnych, zbyt krótkie)

Środowisko:

Chrome wersja 90.0.4430.212

Ubuntu: 20.04 LTS

Warunki wstępne:

1. Uruchomiona przeglądarka
2. Na stronie <https://www.castorama.pl>
3. Użytkownik niezalogowany

Kroki:

1. Zaakceptuj politykę cookies
2. Kliknij Logowanie i rejestracja
3. Wpisz Adres e-mail (w części Rejestracja)
4. Wpisz hasło (w części Rejestracja; brak dużych liter, znaków specjalnych, zbyt krótkie)
5. Powtórz hasło
6. Zaznacz oświadczenie o zapoznaniu z regulaminem
7. Kliknij Załóż konto

Oczekiwany rezultat:

Użytkownik dostaje informację: "Hasło powinno składać się co najmniej z 8 znaków: dużej litery, małej litery, cyfry lub znaku specjalnego (!\'^£\$%&*(){@#~?><>,|=_+~})."

Warunki końcowe:

Konto nie zostaje założone

2. Automatyzacja przypadku testowego przy pomocy Selenium Webdriver:

```
import unittest

from selenium import webdriver

from time import sleep

from selenium.webdriver.common.keys import Keys

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected_conditions

valid_email = "s.dobiech@gmail.com"

invalid_pswd = "haslo"

class Registration(unittest.TestCase):

    def setUp(self):

        self.driver = webdriver.Chrome()

        self.driver.maximize_window()

        self.driver.get("https://www.castorama.pl/")

        self.driver.implicitly_wait(5)

    def testInvalidPassword(self):

        driver=self.driver

        WebDriverWait(self.driver, 5).until(expected_conditions.element_to_be_clickable((By.NAME,"Zaakceptuj")))

        driver.find_element_by_name("Zaakceptuj").click()

        sleep(5)

        driver.find_element_by_xpath('//a[@title="Logowanie i rejestracja"]').click()

        sleep(5)

        email_input = driver.find_element_by_id("email_address")

        email_input.send_keys(valid_email)

        pswd_input = driver.find_element_by_id("password")

        pswd_input.send_keys(invalid_pswd)

        pswd2_input = driver.find_element_by_name("confirmation")

        pswd2_input.send_keys(invalid_pswd)

        driver.find_element_by_xpath('//label[@for="is_accept"]').click()

        sleep(5)
```

```

driver.find_element_by_xpath('//*[@id="form-validate-register"]/button[@type="submit"]').click()

sleep(5)

error_messages = driver.find_elements_by_xpath('//*[@class="validation-advice"'])
visible_error_notices = list()

for error in error_messages:
    if error.is_displayed():
        visible_error_notices.append(error)

self.assertEqual(len(visible_error_notices), 1, msg="Liczba widocznych komunikatow nie zgadza sie!")

self.assertEqual(visible_error_notices[0].text, "Hasło powinno składać się co najmniej z 8 znaków: dużej litery, małej litery,
cyfry lub znaku specjalnego (!\\'^€$%&*(){@#~?><>,|=_+~}).")

def tearDown(self):
    self.driver.quit()

if __name__ == "__main__":
    unittest.main(verbosity=2)

```

3. Uwagi końcowe:

Podczas tworzenia skryptu natrafiono na dosyć ciekawy problem.

Treść komunikatu o błędnym hasle zawiera znak „\”.

W języku Python jest to tzw. biały znak, co ujawniło się następującym błędem:

```

tester@tester-m:~/Desktop/15-16.05$ python3 CastoramaDobiechSylwia.py
testInvalidPassword (__main__.Registration) ... FAIL

=====
FAIL: testInvalidPassword (__main__.Registration)
-----
Traceback (most recent call last):
  File "CastoramaDobiechSylwia.py", line 40, in testInvalidPassword
    self.assertEqual(visible_error_notices[0].text, "Hasło powinno składać się co najmniej z 8 znaków: dużej litery, małej litery, cyfry lub z
naku specjalnego (!\\'^€$%&*(){@#~?><>,|=_+~}).")
AssertionError: "Hasł[72 chars], cyfry lub znaku specjalnego (!\\'^€$%&*(){@#~?><>,|=_+~})." != "Hasł[72 chars], cyfry lub znaku specjalnego
(!\\'^€$%&*(){@#~?><>,|=_+~})."
?
+ Hasło powinno składać się co najmniej z 8 znaków: dużej litery, małej litery, cyfry lub znaku specjalnego (!\\'^€$%&*(){@#~?><>,|=_+~}).

Ran 1 test in 89.448s
FAILED (failures=1)

```

Rozwiązaniem było dopisanie przed wspomnianym „\” drugiego takiego znaku.

Uznano również za słuszne zastosowanie metody, która sprawi, że żaden krok dalej nie zostanie wykonany dopóki nie zaakceptujemy polityki cookies strony. Ponieważ komunikat pojawia się na stronie dosyć szybko, to aby sprawdzić czy kod został napisany prawidłowo przeniesiono polecenie do etapu, w którym strona nie prosi już o akceptację cookies. Zgodnie z oczekiwaniami otrzymano następujący komunikat:

```
tester@tester-m:~/Desktop/15-16.05$ python3 CastoramaDobiechSylwia.py
testInvalidPassword (__main__.Registration) ... ERROR

=====
ERROR: testInvalidPassword (__main__.Registration)
-----
Traceback (most recent call last):
  File "CastoramaDobiechSylwia.py", line 25, in testInvalidPassword
    WebDriverWait(self.driver, 5).until(expected_conditions.element_to_be_clickable((By.NAME,"Zaakceptuj")))
  File "/home/tester/.local/lib/python3.8/site-packages/selenium/webdriver/support/wait.py", line 80, in until
    raise TimeoutException(message, screen, stacktrace)
selenium.common.exceptions.TimeoutException: Message:

-----
Ran 1 test in 36.814s

FAILED (errors=1)
```

Podczas pisania skryptu udało się wykrozystać następujące lokalizatory:

-NAME;

-ID;

-XPATH;

oraz zastosować bardziej rozbudowany XPATH.

W przyszłości skrypt można rozbudować np. o kontrolę poprawności adresu e-mail.