**Automatyzacja przypadku**

**testowego przy pomocy**

**Selenium Webdriver**

**Autor:**

**Irmina Kwiecień**

**Aleksandra Niewęgłowska**

Katowice, 18.04.2018

**I. Przypadek testowy**

**ID: 001**

**Tytuł:** Rejestracja nowego użytkownika używając niepoprawnego hasła do weryfikacji

**Środowisko:** Chrome wersja 65.0.3325.181, PyCharm 2016.2.3

**Warunek wstępny:** Uruchomiona przeglądarka. Użytkownik nie jest zalogowany.

**Kroki:**

1. Wejdź na stronę http://www.sukienkimm.pl/

2. Kliknij "Twoje konto".

3. Kliknij "Załóż konto".

4. Wpisz imię.

5. Wpisz nazwisko.

6. Wpisz w adresie do wysyłki ulicę.

7. Wpisz w adresie do wysyłki numer domu.

8. Wpisz w adresie do wysyłki kod pocztowy.

9. Wpisz w adresie do wysyłki miasto.

10.Wprowadź adres email.

11. Wprowadź hasło.

12. Wprowadź niepoprawne hasło w pole "Powtórz hasło".

13. Zaakceptuj Regulamin.

14. Zaakceptuj Zgodę na przetwarzanie danych osobowych.

15. Kliknij "Rejestruj".

**Oczekiwany rezultat:**

Rejestracja nie powodzi się

Użytkownik dostaje informację, że potwierdzenie hasła musi być takie same jak hasło.

**Kod testu:**

*# -\*- coding: utf-8 -\*-*

import unittest

from time import sleep

from selenium import webdriver

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

valid\_name = "Anna"

valid\_surname = "Nowak"

valid\_street = "Prosta"

valid\_number = "4"

valid\_code = "42-789"

valid\_city = "Cracov"

valid\_email = "anna.nowak@gmail.com"

valid\_password = "asdfzxcv"

valid\_password2 = "asdfqwer"

class test(unittest.TestCase):

def setUp(self):

self.driver = webdriver.Chrome()

self.driver.maximize\_window()

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

def test\_exaple(self):

driver = self.driver

login\_shop=driver.find\_element\_by\_class\_name("loginIco")

login\_shop.click()

sleep(2)

cookies\_info = driver.find\_element\_by\_id("close-cookie-info")

cookies\_info.click()

sleep(2)

regestration\_shop = driver.find\_element\_by\_class\_name('xlarge')

regestration\_shop.click()

sleep(2)

name\_user = driver.find\_element\_by\_id('usr\_first\_name')

name\_user.send\_keys(valid\_name)

sleep(1)

surname\_user = driver.find\_element\_by\_id("usr\_last\_name")

surname\_user.send\_keys(valid\_surname)

sleep(1)

street\_user = driver.find\_element\_by\_id("usr\_address")

street\_user.send\_keys(valid\_street)

sleep(1)

number\_user = driver.find\_element\_by\_id("usr\_address\_str\_no")

number\_user.send\_keys(valid\_number)

sleep(1)

code\_user = driver.find\_element\_by\_id("usr\_post\_code")

code\_user.send\_keys(valid\_code)

sleep(1)

city\_user = driver.find\_element\_by\_id("usr\_city")

city\_user.send\_keys(valid\_city)

sleep(1)

email\_user = driver.find\_element\_by\_id("usr\_email")

email\_user.send\_keys(valid\_email)

sleep(1)

password\_user = driver.find\_element\_by\_id("usr\_password")

password\_user.send\_keys(valid\_password)

sleep(1)

password2\_user = driver.find\_element\_by\_id("usr\_password\_confirm")

password2\_user.send\_keys(valid\_password2)

sleep(1)

agree\_user = driver.find\_element\_by\_xpath('//\*[@id="usr\_terms\_agree"]')

agree\_user.click()

checkbox\_user = driver.find\_element\_by\_xpath('//\*[@id="usr\_data\_manage"]')

checkbox\_user.click()

sleep(2)

registration\_user = driver.find\_element\_by\_id("newUser\_submit\_button")

registration\_user.click()

sleep(3)

WebDriverWait(driver, 3).until(EC.alert\_is\_present(), '')

alert = driver.switch\_to.alert

assert alert.text == u'Potwierdzenie hasła musi być takie same jak hasło !'

alert.accept()

print("alert accepted")

def tearDown(self):

self.driver.quit()

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

**Kod jest dostępny w repozytorium:**

https://github.com/IrminaKwiecien/sklep/blob/master/main.py

**Wnioski:**

Automatyzacja przypadku testowego powiodła się.

**II. Przypadek testowy**

**ID: 002**

**Tytuł:** Próba zrealizowania kuponu rabatowego używając błędnego kodu.

**Środowisko:** Chrome wersja 65.0.3325.181, PyCharm 2016.2.3

**Warunek wstępny:** Uruchomiona przeglądarka.

**Kroki:**

1. Wejdź na stronę: http://www.sukienkimm.pl/sukienki/na-wesele/.

2. Kliknij pierwszą dostępną sukienkę z listy.

3. Kliknij pierwszy dostępny rozmiar.

4. Sprawdź, czy na przycisku “Kup teraz” wyświetla się poprawny rozmiar. Format wyświetlanego tekstu to: “Kup”+ rozmiar +”teraz”.

5. Dodaj sukienkę do koszyka.

6. Wprowadź błędny kod rabatowy.

7. Kliknij “aktywuj kupon”.

**Oczekiwany rezultat:**

Kod rabatowy nie przechodzi weryfikacji.

Użytkownik otrzymuje informację zwrotną o treści, że podany kod rabatowy jest błędny.

**Kod testu:**

*# -\*- coding: utf-8 -\*-*

import unittest

from time import sleep

from selenium import webdriver

invalid\_discount = "A2A2A2A2"

class test(unittest.TestCase):

def setUp(self):

self.driver = webdriver.Chrome()

self.driver.maximize\_window()

self.driver.get("http://www.sukienkimm.pl/sukienki/na-wesele/")

def test\_exaple(self):

driver = self.driver

dresswedding = driver.find\_element\_by\_css\_selector(".pr\_list\_div")

availabledress = dresswedding.find\_elements\_by\_css\_selector('.preview')

pressbutton = availabledress[0].find\_element\_by\_css\_selector('a')

pressbutton.click()

sleep(3)

buttonBar = driver.find\_element\_by\_css\_selector(".rozmiary\_div")

availableSizes = buttonBar.find\_elements\_by\_css\_selector(".atts\_6")

sizeLink = availableSizes[0].find\_element\_by\_css\_selector('a')

sizeLink.click()

sleep(3)

selectedSize = driver.find\_element\_by\_id("selectedSize")

assert selectedSize.text == sizeLink.text

cookies\_info = driver.find\_element\_by\_id("close-cookie-info")

cookies\_info.click()

sleep(1)

addToBasket = driver.find\_element\_by\_id("add2BasketBtn")

addToBasket.click()

sleep(1)

discount =driver.find\_element\_by\_xpath('//\*[@id="pcode"]')

discount.click()

discount.send\_keys(invalid\_discount)

sleep(1)

activateBar = driver.find\_element\_by\_css\_selector(".kupon")

activateBar2 = activateBar.find\_element\_by\_css\_selector(".button")

activateBar2.click()

sleep(5)

modalcontent = driver.find\_element\_by\_css\_selector(".nyroModalIframe")

frames = modalcontent.find\_elements\_by\_css\_selector("iframe")

assert len(frames) == 1

frame = frames[0]

driver.switch\_to.frame(frame)

error = driver.find\_element\_by\_css\_selector("h1")

assert error.text == u"BŁĄD: KOD NIEPOPRAWNY"

sleep(3)

def tearDown(self):

self.driver.quit()

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

**Kod jest dostępny w repozytorium:**

https://github.com/IrminaKwiecien/sklep/blob/master/projekt3.py

**Wnioski:**

Automatyzacja przypadku testowego powiodła się.