

Automatyzacja przypadku testowego przy pomocy Selenium Webdriver

Autor:
Irmina Kwiecień
Aleksandra Niewęglowska

Katowice, 08.05.2018

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 "Załącz 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()
        cookies_info = driver.find_element_by_id("close-cookie-info")
        cookies_info.click()
        sleep(1)
        registration_shop = driver.find_element_by_class_name('xlarge')
        registration_shop.click()
        name_user = driver.find_element_by_id('usr_first_name')
        name_user.send_keys(valid_name)
        surname_user = driver.find_element_by_id("usr_last_name")
        surname_user.send_keys(valid_surname)
        street_user = driver.find_element_by_id("usr_address")
        street_user.send_keys(valid_street)
        number_user = driver.find_element_by_id("usr_address_str_no")
        number_user.send_keys(valid_number)
        code_user = driver.find_element_by_id("usr_post_code")
        code_user.send_keys(valid_code)
        city_user = driver.find_element_by_id("usr_city")
        city_user.send_keys(valid_city)
        email_user = driver.find_element_by_id("usr_email")
        email_user.send_keys(valid_email)
        password_user = driver.find_element_by_id("usr_password")
```

```
password_user.send_keys(valid_password)
password2_user = driver.find_element_by_id("usr_password_confirm")
password2_user.send_keys(valid_password2)
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()
registration_user = driver.find_element_by_id("newUser_submit_button")
registration_user.click()
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()
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()
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")
driver.execute_script("window.scrollTo(0, 30)")
addToBasket.click()
discount = driver.find_element_by_xpath("//*[@id='pcode']")
driver.execute_script("window.scrollTo(0, 300)")
discount.click()
discount.send_keys(invalid_discount)
activateBar = driver.find_element_by_css_selector(".kupon")
activateBar2 = activateBar.find_element_by_css_selector(".button")
activateBar2.click()
sleep(3)
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"

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.