

Automatyzacja przypadku testowego przy pomocy Selenium Webdriver



Autor:

Przemysław Fijak

Tester oprogramowania (zaoczne)

Gliwice 2020/2021

1. Przypadek testowy

ID: 001

Tytuł: Rejestracja nowego użytkownika używając niepoprawnego adresu e-mail

Środowisko: Ubuntu 20.04 LTS, Chrome 90.0.4430.212 (64-bit)

Warunek wstępny: Uruchomiona przeglądarka. Użytkownik nie jest zalogowany.

Kroki:

1. Wejdź na stronę <https://wizzair.com/pl-pl#/>
2. Kliknij „Zaloguj się”
3. Kliknij „Rejestracja”
4. Wpisz imię
5. Wpisz nazwisko
6. Wybierz płeć
7. Wprowadź nr telefonu
8. Wprowadź niepoprawny e-mail
9. Wprowadź hasło
10. Wybierz kraj

Oczekiwany rezultat:

Rejestracja nie powodzi się.

Użytkownik dostaje informację, że wprowadzony e-mail jest niepoprawny.

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
```

```
valid_firstname = "Marian"
valid_lastname = "Nowak"
valid_gender = "male"
valid_country_code = "+48"
valid_phone = "111222333"
valid_password = "Qwerty123@"
valid_country = "Polska"
```

```
invalid_email = "qweasd.pl"
```

```
class WizzairRegistration(unittest.TestCase):
    def setUp(self):
        self.driver = webdriver.Chrome()

        self.driver.maximize_window()
```

```

self.driver.get("https://wizzair.com/pl-pl#/")

self.driver.implicitly_wait(600)

def testInvalidEmail(self):
    driver = self.driver
    zaloguj_btn = driver.find_element_by_xpath('//button[@data-test="navigation-menu-signin"]')
    zaloguj_btn.click()
    sleep(3)

    rejestracja_btn = driver.find_element_by_css_selector('button[data-test="registration"]')
    rejestracja_btn.click()
    sleep(3)

    imie_input = driver.find_element_by_name('firstName')
    imie_input.click()
    imie_input.send_keys(valid_firstname)
    sleep(1)

    nazwisko_input = driver.find_element_by_name('lastName')
    nazwisko_input.click()
    nazwisko_input.send_keys(valid_lastname)
    sleep(1)

    if valid_gender == "male":
        imie_input.click()
        driver.find_element_by_xpath('//label[@data-test="register-gendermale"]').click()
        sleep(1)
    else:
        nazwisko_input.click()
        driver.find_element_by_xpath('//label[@data-test="register-genderfemale"]').click()
        sleep(1)

    driver.find_element_by_xpath('//div[@data-test="booking-register-country-code"]').click()
    cc_input = driver.find_element_by_name('phone-number-country-code')
    cc_input.send_keys(valid_country_code, Keys.RETURN)
    sleep(1)

    phone_input = driver.find_element_by_name('phoneNumberValidDigits')
    phone_input.send_keys(valid_phone)
    sleep(1)

    email_input = driver.find_element_by_xpath('//input[@data-test="booking-register-email"]')
    email_input.click()
    email_input.send_keys(invalid_email)
    sleep(1)

    psswd_input = driver.find_element_by_name('password')
    psswd_input.send_keys(valid_password)
    sleep(1)

    country_input = driver.find_element_by_xpath('//input[@data-test="booking-register-country"]')
    country_input.click()
    countries = driver.find_elements_by_xpath('//div[@class="register-form__country-container__locations"]/label')
    for label in countries:

```

```

    option = label.find_element_by_tag_name('strong')
    if option.get_attribute('innerText') == valid_country:
        option.location_once_scrolled_into_view
        option.click()
        break
    sleep(2)

    error_messages = driver.find_elements_by_xpath('//*[@class="input-error__message"]/span')

    visible_error_notices = list() #pusta lista
    for error in error_messages:
        if error.is_displayed():
            visible_error_notices.append(error)

    assert len(visible_error_notices) == 1, "Liczba widocznych komunikatow nie zgadza sie!"
    self.assertEqual(len(visible_error_notices), 1, msg="Liczba widocznych komunikatow nie zgadza sie!")
    self.assertEqual(visible_error_notices[0].text, "Nieprawidłowy adres e-mail")

    def tearDown(self):
        #zakonczenie testu
        self.driver.quit()

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

```

3. Uwagi końcowe

Automatyzacja przypadku testowego (test funkcjonalny) powiodła się. Test może być wrażliwy na zmianę struktury strony z powodu konieczności stosowania długich ścieżek w lokalizatorach XPATH i CSS Selector.