

Automatyzacja przypadków testowych przy pomocy Selenium WebDriver



Selenium WebDriver

Autorzy:

Patrycja Respondek-Szpakowska, Magdalena Środa,

Ewelina Walczak, Adrian Piechaczek

Katowice, 6.05.2018 r.

I. Przypadki testowe

PRZYPADEK TESTOWY

ID: **001**

TYTUŁ: **Nieprawidłowa rejestracja** (podanie nieodpowiedniego adresu email)

PRIORYTET: Wysoki

WERSJA: 1

AUTOR: WSB Team

AUTOMATYZACJA: Tak

ŚRODOWISKO: Chrome Wersja 64.0.3282.140, Ubuntu 16.04 LTS

WARUNKI WSTĘPNE:

1. Użytkownik jest podłączony do sieci Internet
2. Przeglądarka jest uruchomiona
2. Użytkownik znajduje się na głównej stronie <https://zrabatowani.pl/>

****KROK 1****

Wybierz (kliknij w) zakładkę
"Zarejestruj".

Wyświetlona została strona do rejestracji
wraz z formularzem rejestracyjnym.

****KROK 2****

Wypełnij formularz prawidłowymi
danymi (tylko imię i hasło), np.:
Imię: Agnieszka
hasło: Agnieszka1234
oraz **nieprawidłowym** adresem
email, np.:
email: Agnieszka@ccc

Pola zostały wypełnione. Nie pojawiają się
żadne komunikaty.

****KROK 3****

Zaznacz checkboxa "Akceptuję
regulamin".

Checkbox został zaznaczony.

****KROK 4****

Kliknij przycisk "Wyślij".

Pod polem "email" wyświetlona została
informacja: ""Agnieszka@ccc" nie jest
poprawnym adresem e-mail."
**Użytkownik nie został
zarejestrowany.**

PRZYPADEK TESTOWY

ID: **002**

TYTUŁ: **Nieprawidłowa rejestracja** (podanie nieodpowiedniego hasła)

PRIORYTET: Wysoki

WERSJA: 1

AUTOR: WSB Team

AUTOMATYZACJA: Tak

ŚRODOWISKO: Chrome Wersja 64.0.3282.140, Ubuntu 16.04 LTS

WARUNKI WSTĘPNE:

1. Użytkownik jest podłączony do sieci Internet
2. Przeglądarka jest uruchomiona
2. Użytkownik znajduje się na głównej stronie <https://zrabatowani.pl/>

****KROK 1****

Wybierz (kliknij w) zakładkę
"Zarejestruj".

Wyświetlona została strona do rejestracji
wraz z formularzem rejestracyjnym.

****KROK 2****

Wypełnij formularz prawidłowymi
danymi (tylko imię i email), np.:
Imię: Agnieszka
email: Agnieszka@o2.com
oraz **nieodpowiednim** hasłem, np.:
hasło: 123

Pola zostały wypełnione. Nie pojawiają się
żadne komunikaty.

****KROK 3****

Zaznacz checkboxa "Akceptuję
regulamin".

Checkbox został zaznaczony.

****KROK 4****

Kliknij przycisk "Wyślij".

Pod polem "email" wyświetlona została
informacja: "Podane hasło jest za krótkie.
Wymyśl silniejsze hasło."
**Użytkownik nie został
zarejestrowany.**

PRZYPADEK TESTOWY

ID: **003**

TYTUŁ: **Logowanie użytkownika**

PRIORYTET: Wysoki

WERSJA: 1

AUTOR: WSB Team

AUTOMATYZACJA: Tak

ŚRODOWISKO: Chrome Wersja 64.0.3282.140, Ubuntu 16.04 LTS

WARUNKI WSTĘPNE:

1. Użytkownik jest podłączony do sieci Internet
2. Przeglądarka jest uruchomiona
2. Użytkownik znajduje się na głównej stronie <https://zrabatowani.pl/>

****KROK 1****

Wybierz (kliknij w) zakładkę "Zaloguj".	Wyświetlony zostaje panel do logowania.
---	---

****KROK 2****

Wypełnij panel prawidłowymi danymi (tylko imię i email), np.: email: agula5376@gmail.com hasło: AGULA5376	Pola zostały prawidłowo wypełnione. Nie pojawiają się żadne komunikaty.
---	--

****KROK 3****

Kliknij przycisk "Zaloguj".	Użytkownik jest zalogowany .
-----------------------------	-------------------------------------

****KROK 4****

Sprawdź czy w nagłówku pojawił się tekst powitalny: <i>Witaj Agnieszka</i> .	Tekst wyświetlił się prawidłowo.
--	----------------------------------

II. Automatyzacja przypadków testowych przy pomocy Selenium Webdriver

```
# -*- coding: utf-8 -*-
import sys
reload(sys)
sys.setdefaultencoding('utf-8')
from selenium import webdriver
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
import unittest
```

```
from time import sleep
```

```
url = 'https://zrabortowani.pl/'
logo_css = 'logo-top'
registration_css = 'no_p'
name_field_css = 'register_profile_name'
email_field_css = 'register_email'
password_field_css = 'register_password'
checkbox_css = 'checkbox-label'
register_button_css = 'register-submit'
invalid_info = 'list-unstyled'
alert_css = 'alert-danger'
log_tab_css = 'dropdownLogin'
user_log_email_css = 'username'
user_log_password_css = 'password'
log_button_css = 'div[id=top-menu] button[name=_submit]'
user_check_css = 'user-welcome'
```

```
valid_name = 'Agnieszka'
valid_email = 'agula5376@gmail.com'
valid_email2 = 'Agnieszka@o2.com'
invalid_email = 'Agnieszka@ccc'
valid_password = 'AGULA5376'
invalid_password = '123'
```

```
class ZrabortowaniRegistration(unittest.TestCase):
```

```
    @classmethod
    def setUpClass(self):
        self.driver = webdriver.Chrome()
        self.driver.get(url)

    def setUp(self):
        self.driver.maximize_window()
        register_button = self.driver.find_element_by_class_name(registration_css)
        register_button.click()
        WebDriverWait(self.driver, 5).until(EC.visibility_of_element_located(By.CLASS_NAME, logo_css))

    def test1_invalid_mail(self):
        driver = self.driver
        print("\n" + '***TEST 1 - INCORRECT REGISTRATION - INVALID EMAIL***')
        user_name = driver.find_element_by_id(name_field_css)
        user_name.send_keys(valid_name)
        user_email = driver.find_element_by_id(email_field_css)
        user_email.send_keys(invalid_email)
        user_password = driver.find_element_by_id(password_field_css)
        user_password.send_keys(valid_password)
        checkbox = driver.find_element_by_class_name(checkbox_css)
        checkbox.click()
        register_button = driver.find_element_by_class_name(register_button_css)
        register_button.click()
        WebDriverWait(driver, 5).until(EC.text_to_be_present_in_element((By.CLASS_NAME, invalid_info),
('"' + invalid_email + '"' nie jest poprawnym adresem e-mail.')))

    def test2_invalid_password(self):
        driver = self.driver
        print("\n" + '***TEST 2 - INCORRECT REGISTRATION - INVALID PASSWORD***')
        user_name = driver.find_element_by_id(name_field_css)
        user_name.send_keys(valid_name)
        user_email = driver.find_element_by_id(email_field_css)
        user_email.send_keys(valid_email2)
        user_password = driver.find_element_by_id(password_field_css)
        user_password.send_keys(invalid_password)
        checkbox = driver.find_element_by_class_name(checkbox_css)
        checkbox.click()
        register_button = driver.find_element_by_class_name(register_button_css)
        register_button.click()
        WebDriverWait(driver, 5).until(EC.text_to_be_present_in_element((By.CLASS_NAME, invalid_info),
('Podane hasło jest za krótkie. Wymyśl silniejsze hasło.')))
```

```

def test3_login(self):
    driver = self.driver
    print("\n" + '***TEST 3 - CORRECT_LOGIN***')
    log_tab = driver.find_element_by_id(log_tab_css)
    log_tab.click()
    user_email = driver.find_element_by_id(user_log_email_css)
    user_email.send_keys(valid_email)
    user_password = driver.find_element_by_id(user_log_password_css)
    user_password.send_keys(valid_password)
    log_button = driver.find_element_by_css_selector(log_button_css)
    log_button.click()
    WebDriverWait(driver, 5).until(EC.text_to_be_present_in_element((By.CLASS_NAME, user_check_css),
("Witaj " + valid_name)))

def tearDown(self):
    pass

@classmethod
def tearDownClass(self):
    self.driver.close()

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

```

Kod jest dostępny w repozytorium:

<https://github.com/elciak82/mikroprojekt-selenium> w katalogu
testy/testyProjekt.py

III. Uwagi końcowe¹

Automatyzacja przypadków testowych powiodła się. Testy automatyczne wykorzystują selektory CSS, głównie *id* i *className*.

Zgodnie z przyjętymi standardami atrybut *id* jest unikalny z tego też względu jest on najlepszym lokatorem. Dobre praktyki wskazują by używać go zawsze wtedy, kiedy występuje taka możliwość. Zastosowano go wszędzie tam, gdzie była taka możliwość.

Jedną z alternatyw, gdy nie posiadamy unikalnego atrybutu *id* jest zastosowanie atrybutu *className*. Klasa nie musi być unikalna w wybranym dokumencie, ale jest polecana ze względu na ogromną łatwość odszyfrowania, co się pod nią kryje. Należy być jednak ostrożnym, ponieważ nawet jeżeli klasa jest obecna w danej chwili to w przyszłości może zostać dodany kolejny element z taką samą klasą. Może to skutkować problemami w naszych testach. Dlatego też warto podczas budowy strony stosować unikalne selektory *id*, dzięki temu testy są prostsze w utrzymaniu.

¹ Wykorzystano informacje ze stron: <https://devstyle.pl/2017/06/12/nobody-expects-the-spanish-inquisition-czyli-testy-ui/> oraz <http://testerzy.pl/baza-wiedzy/akcja-automatyzacja-czesc-5-klasa-webelement>.