

Automatyzacja przypadku testowego przy pomocy Selenium Webdriver



I. Przypadek testowy

ID: 001

Tytuł: Rejestracja nowego użytkownika używając nieakceptowanej daty urodzenia.

Środowisko: Chromium Wersja 83.0.4103.61, Manjaro Linux 20.0.3

Warunek wstępny: Uruchomiona przeglądarka. Użytkownik nie jest zalogowany.

Dodatkowe informacje: Data nieakceptowana to data urodzenia wcześniejsza niż 1900 r.

Kroki:

1. Wejść na stronę "<https://www.biblionetka.pl/>"
2. Kliknij w prawym górnym rogu ZAREJESTRUJ SIĘ
3. Wprowadź login
4. Wprowadź hasło
5. Potwierdź hasło
6. Wprowadź email
7. Powtórz email
8. Wprowadź datę urodzenia
9. Wprowadź płeć
10. Zaakceptuj regulamin.

Oczekiwany rezultat:

Rejestracja nie powodzi się.

Użytkownik dostaje informację: „Niepoprawny rok urodzenia”.

II. Automatyzacja przypadku testowego przy pomocy Selenium Webdriver

```
import unittest
from selenium import webdriver
from time import sleep
from faker import Faker

""" 'pl_PL' jest niezbędne aby skorzystać z generowania danych do testów wykorzystujących np. polski
numer nip"""
fake = Faker(['en_US', 'pl_PL'])

valid_first_name = fake.first_name()
valid_last_name = fake.last_name()
valid_password = fake.password(length=5)
valid_email = fake.free_email()
invalid_date = str(fake.date_between(start_date='-121y', end_date='-121y'))

class BiblionetkaRegistration(unittest.TestCase):
    """
    Scenariusz testowy: Rejestracja nowego użytkownika na stronie www.biblionetka.pl/
    """

    def setUp(self):
        """
        Warunki wstępne:
        Przeglądarka otwarta na https://www.biblionetka.pl/
        """
        self.driver = webdriver.Chrome()
        self.driver.maximize_window()
        self.driver.get("https://www.biblionetka.pl/")
        self.driver.implicitly_wait(10)

    def tearDown(self):
        """ Sprzątanie po teście """
        self.driver.quit()

    def test_invalid_date_of_birth(self):
        """
        Rejestracja nowego użytkownika
        używając nieakceptowalnego roku urodzenia - dane niepoprawne
        (data urodzenia wcześniejsza niż rok 1900)
        """
        driver = self.driver
        # KROKI:

        """
        Wyłączam komunikat o ciasteczkach - "zasłania" on dolne pola formularza, przez co Selenium ich
        "nie widzi"
        """
```

Sposób bez użycia javascript:

```
driver.find_element_by_xpath('//*[@id="ctl00_MCP_CUW___CustomNav0_StepNextButton"]').send_keys(Keys.ENTER)
```

```
"""
```

```
# Wyłączam komunikat o ciasteczkach z użyciem javascript
```

```
accept_cookies = driver.find_element_by_xpath('//*[@id="accept-cookies-checkbox"]')
```

```
driver.execute_script("arguments[0].click();", accept_cookies)
```

```
# 1. Kliknij w prawym górnym rogu ZAREJESTRUJ SIĘ
```

```
rejestracja_button = driver.find_element_by_id('ctl00_registrationLink')
```

```
rejestracja_button.click()
```

```
# 2. Wprowadź login
```

```
login_field =
```

```
driver.find_element_by_name("ctl00$MCP$CUW$CreateUserStepContainer$UserName")
```

```
login_field.send_keys(valid_first_name + valid_last_name)
```

```
# 3. Wprowadź hasło
```

```
# password_field = driver.find_element_by_xpath('//*[@placeholder="xxxx"]')
```

```
password_field =
```

```
driver.find_element_by_xpath('//*[@id="ctl00_MCP_CUW_CreateUserStepContainer_Password"]')
```

```
password_field.send_keys(valid_password)
```

```
# 4. Potwierdź hasło
```

```
# poprawa czytelności - rozbiecie jednej linii kodu na dwie, aby długość linii nie przekraczała 80 znaków),
```

```
# dla odmiany bez użycia dodatkowej zmiennej
```

```
driver.find_element_by_xpath \
```

```
('//*[ @ id =
```

```
"ctl00_MCP_CUW_CreateUserStepContainer_ConfirmPassword"]').send_keys(valid_password)
```

```
# 5. Wprowadź email
```

```
driver.find_element_by_name("ctl00$MCP$CUW$CreateUserStepContainer$Email").send_keys(valid_email)
```

```
# 6. Powtórz email
```

```
driver.find_element_by_name("ctl00$MCP$CUW$CreateUserStepContainer$Email2").send_keys(valid_email)
```

```
# 7. Wprowadź datę urodzenia
```

```
driver.find_element_by_name("ctl00$MCP$CUW$CreateUserStepContainer$BirthYear").send_keys(invalid_date)
```

8. Wprowadź płeć

```
driver.find_element_by_xpath('//*[@id="ctl00_MCP_CUW_CreateUserStepContainer_Sex_1"]').click()
```

9. Zaakceptuj regulamin

"""

W sytuacji jeśli komunikat o ciasteczkach nie byłyby wyłączony (lina 52), można wymusić akceptację regulaminu przez "wysłanie" spacji

```
driver.find_element_by_xpath('//*[@id="ctl00_MCP_CUW_CreateUserStepContainer_RulesAccepted"]').send_keys(' ')
"""
```

```
driver.find_element_by_xpath('//*[@id="ctl00_MCP_CUW_CreateUserStepContainer_RulesAccepted"]').click()
```

"""

Kliknięcie rejestracji - nie używam, ponieważ nie chcę dokonać rzeczywistej rejestracji i generować

pustych kont Biblionetce

```
driver.find_element_by_xpath('//*[@ @ id = "ctl00_MCP_CUW___CustomNav0_StepNextButton"]').click()
"""
```

Wizualnie oceniam test"

sleep(10)

"""

Test: sprawdzam oczekiwany rezultat

"""

Wyszukuję wszystkie błędy

```
error_notices = driver.find_elements_by_xpath(
    '//*[@id="ctl00_MCP_CUW_CreateUserStepContainer_RangeValidator1"]')
```

Zapisuję widoczne błędy do listy visible_error_notices

Tworzę pustą listę

```
visible_error_notices = []
```

```
for error in error_notices:
```

```
    # Jesli błąd jest widoczny, dodaj go do listy
```

```
    if error.is_displayed():
```

```
        visible_error_notices.append(error)
```

Sprawdzam, czy jest widoczny tylko jeden błąd

```
assert len(visible_error_notices) == 1
```

Sprawdzam treść widocznego błędu

```
error_text = visible_error_notices[0].get_attribute("innerText")
```

```
assert error_text == "* Niepoprawny rok urodzenia"
```

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

III. Uwagi końcowe

Automatyzacja przypadku testowego (test funkcjonalny) powiodła się.