

STUDIA PODYPLOMOWE „Tester oprogramowania”

Wyższa Szkoła Bankowa w Chorzowie

Automatyzacja przypadku testowego przy pomocy Selenium Webdriver

Autor:

Katarzyna Lecybyl-Szedel

Katowice 2020

I. Przypadek testowy

Tytuł: Wyszukaj lek na stronie <https://ktomalek.pl>

Przypadek testowy: Błędna nazwa leku

Środowisko: Chrome wersja 80.0.3987.163, Windows 10 Pro

Warunki wstępne:

1. Uruchomiona przeglądarka.
2. Otwarta strona <https://ktomalek.pl>

Kroki:

1. Wybierz „1” – „Podaj lokalizację” / „Wybrana lokalizacja”
2. Wpisz nazwę miasta i ulicy
3. Naciśnij klawisz ENTER
4. Wpisz błędną nazwę leku
5. Naciśnij klawisz ENTER

Oczekiwany rezultat:

Wyszukanie leku nie powodzi się.

Użytkownik otrzymuje informację, o braku wyników wyszukiwania: Brak wyników wyszukiwania.

II. Automatyzacja przypadku testowego przy pomocy Selenium Webdriver

```
import unittest
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

# Dane testowe
city_name = 'Katowice'
street_name = 'Wyszyńskiego 7'
drug_name = 'ketonal'
error_drug_info = 'Brak wyników wyszukiwania.'
```

```
class KtoMaLekFindDrugTests(unittest.TestCase):
    """
    Scenariusz testowy: Wyszukanie na stronie https://ktomalek.pl we wskazanej lokalizacji leku według
    zadanego wzorca
    """

    @classmethod
    def setUpClass(cls):
        cls.driver = webdriver.Chrome(executable_path=r"C:\WSB_PROJECTS\chromedriver.exe")
        cls.driver.maximize_window()
        cls.driver.get('https://ktomalek.pl/')

    @classmethod
    def tearDownClass(cls):
        cls.driver.quit()

    def test_find_drug_in_localization(self):
        # Lokalizatory
        change_address_xpath = '//*[@id="krok_1_linked"]/a'
        address_input_box_xpath = '//*[@id="searchAdresu"]'
        search_drug_input_xpath = '//*[@id="search"]'
        drugs_name_list_xpath = '//*[@class="toggle-box hover-box"]//h2'
        no_drugs_xpath = '//*[@id="brakLekowMessage"]'

        driver = self.driver

        # Kliknij w "1" - "Podaj lokalizację" / "Wybrana lokalizacja"
        # Zaczekaj max. 30 s. aż pole będzie aktywne
```

```

change_localization = WebDriverWait(driver, 30).until(
    EC.element_to_be_clickable((By.XPATH, change_address_xpath)),
    f'Change address localization site on page: {driver.current_url} not found')
change_localization.click()

# Kliknij w pole "Wpisz miasto i ulicę"; zaczekaj aż pole będzie aktywne"
enter_address = WebDriverWait(driver, 30).until(
    EC.element_to_be_clickable((By.XPATH, address_input_box_xpath)),
    f'Input address box on page: {driver.current_url} not found')
# Wprowadź adres
address_name = city_name + ', ' + street_name
enter_address.send_keys(address_name)
enter_address.send_keys(Keys.ENTER)

# Zaczekaj aż uaktywni się pole do wprowadzania leku i wprowadź nazwę poszukiwanego leku
enter_drug = WebDriverWait(driver, 30).until(
    EC.element_to_be_clickable((By.XPATH, search_drug_input_xpath)),
    f'Input drug box on page: {driver.current_url} not found')
enter_drug.send_keys(drug_name)
enter_drug.send_keys(Keys.ENTER)

# Zaczekaj aż strona zwróci listę wyszukanych pozycji
# Jeżeli lek nie zostanie odnaleziony zwróć komunikat o braku leku
try:
    drugs_list = WebDriverWait(driver, 20).until(
        EC.presence_of_all_elements_located((By.XPATH, drugs_name_list_xpath)),
        f'Drug \'{drug_name}\'' on page: {driver.current_url} not found')

except:
    error_notice = driver.find_element_by_xpath(no_drugs_xpath)
    print(error_notice.text)
    self.assertEqual(error_notice.text, error_drug_info)

else:
    # Przejdź przez każdy znaleziony element z listy i zweryfikuj czy nazwa poszukiwanego leku
    # zawiera się w nazwie z każdej ze znalezionych pozycji;
    # wypisz numer i nazwę znalezionej pozycji
    for i in range(len(drugs_list)):
        drug_name_text = drugs_list[i].get_attribute("textContent")
        print(f"{i + 1} drug's name: {drug_name_text}")
        with self.subTest(drug_name_text):
            self.assertIn(drug_name.upper(), drug_name_text,
                f'Drug name {drug_name} not contain in {drug_name_text}')

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

```

III. Uwagi końcowe

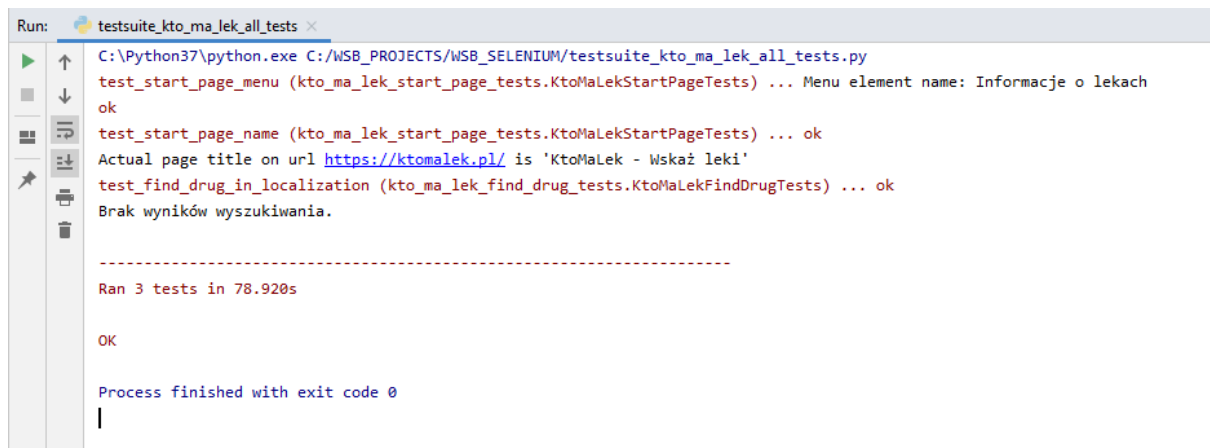
Automatyzacja przypadku testowego powiodła się.

W przykładzie zaimplementowany został wariant dla przypadku testowego: Prawidłowa nazwa leku.

Załączony projekt zawiera następujące pliki:

1. *kto_ma_lek_find_drug_tests.py* – z przedstawioną w dokumencie klasą: `KtoMaLekFindDrugTests(unittest.TestCase)` i testem: `test_find_drug_in_localization(self)`
2. *kto_ma_lek_start_page_tests.py* – z zaimplementowaną klasą: `KtoMaLekStartPageTests(unittest.TestCase)` i testami: `test_start_page_name(self)` oraz `test_start_page_menu(self)`
3. *testsuite_kto_ma_lek_all_tests.py* – ‘runner-a’ dla wszystkich testów.

IV. Wynik testów



The screenshot shows the output of a Selenium test suite execution. The window title is 'Run: testsuite_kto_ma_lek_all_tests'. The output text is as follows:

```
C:\Python37\python.exe C:/WSB_PROJECTS/WSB_SELENIUM/testsuite_kto_ma_lek_all_tests.py
test_start_page_menu (kto_ma_lek_start_page_tests.KtoMaLekStartPageTests) ... Menu element name: Informacje o lekach
ok
test_start_page_name (kto_ma_lek_start_page_tests.KtoMaLekStartPageTests) ... ok
Actual page title on url https://ktomalek.pl/ is 'KtoMaLek - Wskaż leki'
test_find_drug_in_localization (kto_ma_lek_find_drug_tests.KtoMaLekFindDrugTests) ... ok
Brak wyników wyszukiwania.

-----
Ran 3 tests in 78.920s

OK

Process finished with exit code 0
|
```