

**Automatyzacja testów
witryny taniaksiazka.pl
przy pomocy Selenium Webdriver
oraz języka Python.**



Autor:

Katarzyna Gibert

Projekt:

taniaksiazka.pl

Środowisko:

Chrome wersja 111.0.5563.146, Ubuntu 20.04

Scenariusz testowy:

Logowanie użytkownika

Id	Przypadek testowy	Opis przypadku	Krok testu	Opis kroku	Oczekiwany rezultat	Aktualny rezultat	Status
001	Logowanie z prawidłowymi danymi	Klient powinien mieć możliwość zalogowania się	Warunek wstępny	Otwarta strona główna. Użytkownik nie jest zalogowany i posiada konto na stronie			
			Krok 1	Kliknij przycisk "Zaloguj się"	Wyświetli się podstrona z możliwością zalogowania się	Wyświetliła się podstrona z możliwością zalogowania się	Pozytywny/Pass
			Krok 2	Wpisz prawidłowy e-mail w polu "e-mail" w formularz logowania	E-mail został wpisany	E-mail został wpisany	Pozytywny/Pass
			Krok 3	Wpisz prawidłowe hasło w polu "hasło" w formularzu logowania	Hasło zostało wpisane	Hasło zostało wpisane	Pozytywny/Pass
			Krok 4	Kliknij przycisk "Zaloguj się"	Użytkownik został zalogowany	Użytkownik został zalogowany	Pozytywny/Pass
002	Logowanie z nie - prawidłowym hasłem	Użytkownik nie powinien mieć możliwość zalogowania się	Warunek wstępny	Otwarta strona główna. Użytkownik nie jest zalogowany i posiada konto na stronie.			
			Krok 1	Kliknij przycisk "Zaloguj się"	Wyświetli się podstrona z możliwością zalogowania się	Wyświetliła się podstrona z możliwością zalogowania się	Pozytywny/Pass
			Krok 2	Wpisz prawidłowy e-mail w polu "e-mail" w formularz logowania	E-mail został wpisany	E-mail został wpisany	Pozytywny/Pass

			Krok 3	Wpisz nieprawidłowe hasło w polu "hasło" w formularzu logowania	Hasło zostało wpisane	Hasło zostało wpisane	Pozytywny/Pass
			Krok 4	Kliknij przycisk "Zaloguj się"	Użytkownik nie został zalogowany. Wyświetlił się komunikat "Niepoprawne hasło"	Użytkownik nie został zalogowany. Wyświetlił się komunikat "Niepoprawne hasło"	Pozytywny/Pass

Automatyzacja przypadków testowych

```
import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By
from time import sleep
```

```
# DANE TESTOWE
```

```
email = "kasia.gibert@gmail.com"
```

```
password = "kasiagibert"
```

```
invalid_password = "kas"
```

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

```
#Logowanie uzytkownika
```

```
def setUp(self):
```

```
    """ Przygotowanie testu """
```

```
    # 1. Otwarta strona główna
```

```
    self.driver = webdriver.Chrome()
```

```
    self.driver.get("https://www.taniaksiazka.pl/")
```

```
    self.driver.maximize_window()
```

```
    self.driver.implicitly_wait(20)
```

```
    # 2. Użytkownik niezalogowany
```

```
def testCorrectLogin(self):
```

```
    """ Logowanie z prawidłowymi danymi """
```

```
    # 1. Kliknij przycisk „Zaloguj się”
```

```
    log_in = self.driver.find_element(By.XPATH, '//*[@id="user-box"]/div[1]/p/a/strong')
```

```
    log_in.click()
```

```
    # 2. Wpisz prawidłowy e-mail
```

```
    email_input = self.driver.find_element(By.ID, 'loginEmail')
```

```
email_input.send_keys(email)
```

```
# 3. Wpisz haslo
```

```
password_input = self.driver.find_element(By.ID, 'loginPassword')
```

```
password_input.send_keys(password)
```

```
# 4. Kliknij przycisk "Zaloguj sie"
```

```
log_in_button = self.driver.find_element(By.XPATH, '//*[@id="LoginPage"]/form[1]/p/button')
```

```
log_in_button.submit()
```

```
# Testy sprawdzające poprawność adresu URL
```

```
URL = self.driver.current_url
```

```
self.assertEqual("https://www.taniaksiazka.pl/Konto", URL, "Incorrect URL")
```

```
def testInvalidPassword(self):
```

```
    """ Logowanie z nieprawidłowym hasłem """
```

```
# 1. Kliknij przycisk „Zaloguj sie”
```

```
log_in = self.driver.find_element(By.XPATH, '//*[@id="user-box"]/div[1]/p/a/strong')
```

```
log_in.click()
```

```
# 2. Wpisz email
```

```
email_input = self.driver.find_element(By.ID, 'loginEmail')
```

```
email_input.send_keys(email)
```

```
# 3. Wpisz nieprawidłowe hasło
```

```
password_input = self.driver.find_element(By.ID, 'loginPassword')
```

```
password_input.send_keys(invalid_password)
```

```
# 4. Kliknij przycisk "Zaloguj sie"
```

```
log_in_button = self.driver.find_element(By.XPATH, '//*[@id="LoginPage"]/form[1]/p/button')
```

```
log_in_button.submit()
```

```
# Testy sprawdzające poprawność adresu URL
```

```
URL = self.driver.current_url
```

```
self.assertEqual("https://www.taniaksiazka.pl/Logowanie", URL, "Incorrect URL")
```

```
def tearDown(self):
```

```
    self.driver.quit()
```

Scenariusz testowy:

Rejestracja nowego użytkownika

001	Rejestracja nowego użytkownika używając niepoprawnego hasła	Użytkownik nie powinien mieć możliwości zarejestrowania się	Warunek wstępny	Otwarta strona główna. Użytkownik nie posiada konta na stronie. Hasło krótsze niż 4 znaki.			
			Krok 1	Kliknij przycisk "Zaloguj się"	Wyświetli się podstrona z możliwością zarejestrowania się.	Wyświetliła się podstrona z możliwością zarejestrowania się.	Pozytywny /Pass
			Krok 2	Wpisz prawidłowe imię w polu "Imię" w formularzu rejestracji	Imię zostało wpisane	Imię zostało wpisane	Pozytywny /Pass
			Krok 3	Wpisz prawidłowe nazwisko w polu "Nazwisko" w formularzu rejestracji	Nazwisko zostało wpisane	Nazwisko zostało wpisane	Pozytywny /Pass
			Krok 4	Wpisz prawidłowy numer telefonu w polu "Telefon" w formularzu rejestracji	Numer został wpisany	Numer został wpisany	Pozytywny /Pass
			Krok 5	Wpisz prawidłowy e-mail w polu "Twój e-mail" w formularzu rejestracji	E-mail został wpisany	E-mail został wpisany	Pozytywny /Pass
			Krok 6	Wpisz błędne hasło w polu "Hasło" w formularzu rejestracji	Błędne hasło zostało wpisane	Błędne hasło zostało wpisane	Pozytywny /Pass
			Krok 7	Wpisz błędne hasło w polu "Powtórz hasło" w formularzu rejestracji	Błędne hasło zostało wpisane	Błędne hasło zostało wpisane	Pozytywny /Pass
			Krok 8	Zaznacz checkboxa	Checkbox został zaznaczony	Checkbox został zaznaczony	Pozytywny /Pass

			Krok 9	Kliknij przycisk "załóż konto"	Konto nie zostało założone. Wyświetli się komunikat przy polu hasło "Wpisz min. 4 znaki"	Konto nie zostało założone. Wyświetlił się komunikat przy polu hasło "Wpisz min. 4 znaki"	Pozytywny /Pass
002	Rejestracja nowego użytkownika w przypadku nie zaznaczenia checkboxa	Użytkownik nie powinien mieć możliwości zarejestrowania się	Warunek wstępny	Otwarta strona główna. Użytkownik nie posiada konta na stronie.			
			Krok 1	Kliknij przycisk "Zaloguj się"	Wyświetli się podstrona z możliwością zarejestrowania się.	Wyświetliła się podstrona z możliwością zarejestrowania się.	Pozytywny /Pass
			Krok 2	Wpisz prawidłowe imię w polu "Imię" w formularzu rejestracji	Imię zostało wpisane	Imię zostało wpisane	Pozytywny /Pass
			Krok 3	Wpisz prawidłowe nazwisko w polu "Nazwisko" w formularzu rejestracji	Nazwisko zostało wpisane	Nazwisko zostało wpisane	Pozytywny /Pass
			Krok 4	Wpisz prawidłowy numer telefonu w polu "Telefon" w formularzu rejestracji	Numer został wpisany	Numer został wpisany	Pozytywny /Pass
			Krok 5	Wpisz prawidłowy e-mail w polu "Twój e-mail" w formularzu rejestracji	E-mail został wpisany	E-mail został wpisany	Pozytywny /Pass
			Krok 6	Wpisz błędne hasło w polu "Hasło" w formularzu rejestracji	Błędne hasło zostało wpisane	Błędne hasło zostało wpisane	Pozytywny /Pass
			Krok 7	Wpisz poprawne hasło w polu "Powtórz hasło" w formularzu rejestracji	Hasło zostało wpisane	Hasło zostało wpisane	Pozytywny /Pass

			Krok 8	Kliknij przycisk "załóż konto"	Konto nie zostało założone. Wyświetli się komunikat przy polu hasło "To pole jest wymagane"	Konto nie zostało założone. Wyświetlił się komunikat przy polu hasło "To pole jest wymagane "	Pozytywny /Pass
--	--	--	--------	-----------------------------------	---	---	--------------------

Automatyzacja przypadków testowych

```
import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By
from time import sleep
from selenium.webdriver.support.relative_locator import locate_with
```

```
# DANE TESTOWE
```

```
name = "Katarzyna"
```

```
last_name = "Owerko"
```

```
phone_number = "723922682"
```

```
email = "kasia.g@gmail.com"
```

```
password = "kasiagibert"
```

```
invalid_password = "kas"
```

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

```
# Rejestracja nowego uzytkownika
```

```
def setUp(self):
```

```
    """Przygotowanie testu"""
```

```
# 1. Otwarta strona główna
```

```
self.driver = webdriver.Chrome()
```

```
self.driver.get("https://www.taniaksiazka.pl/")
```

```
self.driver.maximize_window()
```

```
self.driver.implicitly_wait(20)
```

```
# 2. Użytkownik niezalogowany
```

```
def testInvalidPassword(self):
```

```
    """ Rejestracja nowego uzytkownika uzywajac niepoprznego hasla """
```

```
# 1. Kliknij przycisk „Zaloguj sie”
```

```
log_in = self.driver.find_element(By.XPATH, '//*[@id="user-box"]/div[1]/p/a/strong')
```

```
log_in.click()
```

2. Wpisz imię

```
name_input = self.driver.find_element(By.ID, 'registerFirstName')
name_input.send_keys(name)
```

3. Wpisz nazwisko

```
last_name_input = self.driver.find_element(By.ID, 'registerLastName')
last_name_input.send_keys(last_name)
```

4. Wpisz numer telefonu

```
phone_number_input = self.driver.find_element(By.ID, 'registerPhone')
phone_number_input.send_keys(phone_number)
```

5. Wpisz e-mail

```
email_input = self.driver.find_element(By.ID, 'registerEmail')
email_input.send_keys(email)
```

6. Wpisz błędne hasło

```
password_input = self.driver.find_element(By.ID, 'registerPassword')
password_input.send_keys(invalid_password)
```

7. Wpisz błędne hasło

```
password1_input = self.driver.find_element(By.ID, 'registerPassword1')
password1_input.send_keys(invalid_password)
```

8. Zaznacz checkboxa

```
checkbox = self.driver.find_element(By.ID, 'registerAgree')
checkbox.click()
```

9. Kliknij załóż konto

```
create_account_button = self.driver.find_element(By.ID, 'CreateAccountSubmit')
create_account_button.submit()
```

Screenshot

```
screenshot = self.driver.get_screenshot_as_file('screen.png')
```

Testy sprawdzające

1. Szukam wszystkich komunikatów o błędzie

```
password_error = self.driver.find_elements(By.CLASS_NAME, 'error')
```

2. Sprawdzam, czy jest jeden taki komunikat

```
self.assertEqual(1, len(password_error))
```

3. Sprawdzam poprawność treści komunikatu i jego widoczność

```
self.assertEqual("Wpisz min. 4 znaki", password_error[0].text, "Komunikat o błędzie ma niepoprawną treść")
```

4. Sprawdzam położenie tego napisu

a) Ponownie szukam tego elementu względem pola hasła

```
password_error_locator = locate_with(By.CLASS_NAME, 'error').near({By.ID:
'registerPassword'})
```

```
password_error_location = self.driver.find_element(password_error_locator)
```

b) Sprawdzam, czy element wyszukany na oba sposoby to w istocie ten sam element


```
self.assertEqual(password_error[0].id, password_error_location.id)
```

```
def testNoCheckbox(self):
```

```
    """ Rejestracja nowego uzytkownika w przypadku nie zaznaczenia checkboxa """
```

```
    # 1. Kliknij przycisk „Zaloguj sie”
```

```
    log_in = self.driver.find_element(By.XPATH, '//*[@id="user-box"]/div[1]/p/a/strong')
```

```
    log_in.click()
```

```
    # 2. Wpisz imię
```

```
    name_input = self.driver.find_element(By.ID, 'registerFirstName')
```

```
    name_input.send_keys(name)
```

```
    # 3. Wpisz nazwisko
```

```
    last_name_input = self.driver.find_element(By.ID, 'registerLastName')
```

```
    last_name_input.send_keys(last_name)
```

```
    # 4. Wpisz numer telefonu
```

```
    phone_number_input = self.driver.find_element(By.ID, 'registerPhone')
```

```
    phone_number_input.send_keys(phone_number)
```

```
    # 5. Wpisz e-mail
```

```
    email_input = self.driver.find_element(By.ID, 'registerEmail')
```

```
    email_input.send_keys(email)
```

```
    # 6. Wpisz hasło
```

```
    password_input = self.driver.find_element(By.ID, 'registerPassword')
```

```
    password_input.send_keys(password)
```

```
    # 7. Wpisz hasło
```

```
    password1_input = self.driver.find_element(By.ID, 'registerPassword1')
```

```
    password1_input.send_keys(password)
```

```
    # 9. Kliknij przycisk "zaloguj konto"
```

```
    create_account_button = self.driver.find_element(By.ID, 'CreateAccountSubmit')
```

```
    create_account_button.submit()
```

```
    # Screenshot
```

```
    screenshot = self.driver.get_screenshot_as_file('screen1.png')
```

```
def tearDown(self):
```

```
    self.driver.quit()
```

Scenariusz testowy:

Sprawdzenie funkcjonalności wyszukiwarki

001	Wyszukiwanie książek zawierających określoną frazę	Użytkownik powinien mieć możliwości wyszukania książki po określonej frazie	Warunek wstępny	Otwarta strona główna. Księgarnia posiada w sprzedaży książkę zawierającą w tytule "sapiens"			
			Krok 1	Kliknij w pole wyszukiwania	Użytkownik klika w pole wyszukiwania	Użytkownik kliknął w pole wyszukiwania	Pozytywny /Pass
			Krok 2	Sprawdź, czy pole wyszukiwania jest puste. Jeżeli nie jest wyczyść je.	Użytkownik sprawdził, czy pole wyszukiwania jest puste. Jeżeli nie było wyczyścił je.	Użytkownik sprawdził, czy pole wyszukiwania jest puste. Jeżeli nie było wyczyścił je.	Pozytywny /Pass
			Krok 3	Wpisz w polu wyszukiwania frazę "sapiens"	Fraza "sapiens" została wpisana w pole wyszukiwania	Fraza "sapiens" została wpisana w pole wyszukiwania	Pozytywny /Pass
			Krok 4	Kliknij przycisk wyszukiwania	Wyświetli się podstrona z listą książek zawierającą w tytule frazę "sapiens"	Wyświetliła się podstrona z listą książek zawierającą w tytule frazę "sapiens"	Pozytywny /Pass

Automatyzacja przypadków testowych

```
import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By
from time import sleep
```

```
# DANE TESTOWE
book = "sapiens"
```

```
class SearchTest(unittest.TestCase):  
# Sprawdzenie funkcjonalności wyszukiwarki
```

```
def setUp(self):  
    """Przygotowanie testu"""
```

```
# 1. Otwarta strona główna  
self.driver = webdriver.Chrome()  
self.driver.get("https://www.taniaksiążka.pl/")  
self.driver.maximize_window()  
self.driver.implicitly_wait(20)
```

```
def testSearchFunction(self):  
    """ Wyszukiwanie książek zawierających określoną frazę """
```

```
# 1. Kliknij w pole wyszukiwania  
search_field = self.driver.find_element(By.ID, 'inputSearch')
```

```
# 2. Sprawdź, czy pole wyszukiwania jest puste. Jeżeli nie wyczyść je  
if search_field == "":  
    print("Pole jest puste")  
else:  
    print("Pole nie jest puste, wyczyść je")  
    search_field.clear()
```

```
# 3. Wpisz w pole wyszukiwania frazę "sapiens"  
search_field.send_keys(book)
```

```
# 4. Kliknij przycisk wyszukiwania  
search_button = self.driver.find_element(By.XPATH, '//*[@id="serach  
form"]/fieldset/button')  
search_button.submit()
```

```
# Sprawdzam poprawność adresu URL  
URL = self.driver.current_url  
self.assertEqual("https://www.taniaksiążka.pl/Szukaj/q-sapiens", URL, "Incorrect URL")
```

```
# Pobranie nazw książek z wyników wyszukiwania  
books = self.driver.find_elements(By.XPATH, "//a[contains(@class, 'ecommerce-datalayer  
product-title ')]")  
books_name = [book.get_attribute("textContent") for book in books]  
for name in books_name:  
    print("Book name:" + name)
```

```
# Pobieranie cen książek z wyników wyszukiwania  
prices = self.driver.find_elements(By.XPATH, "//span[contains(@class, 'product-price')]")  
price_values = [price.get_attribute("textContent") for price in prices]  
for price in price_values:  
    print("Price:" + price)
```

```
# Sprawdzanie pierwszych 3 cen hoteli  
assert price_values[0] == "28,07 zł"  
assert price_values[1] == "38,93 zł"  
assert price_values[2] == "27,95 zł"
```

```
def tearDown(self):  
    self.driver.quit()
```

Uwagi końcowe:

Automatyzacja przypadków testowych powiodła się.