

Testování softwaru (KI/TSW)

Pavel Beránek

1 Dubna, 2025

7. Testování webového UI

7.1. Selenium

Selenium je populární framework pro automatizaci webových prohlížečů, který umožňuje psát testovací skripty simulující chování uživatele – klikání, vyplňování formulářů, přihlašování apod. Lze pomocí něj testovat práci s různými prohlížeči jako je: Chrome, Firefox, Edge atd. Selenium existuje pro mnoho jazyků včetně C#, Ruby, Java a Python.

7.2. Příprava prostředí

Selenium si nainstalujete následujícím příkazem:

```
pip install selenium
```

Kromě toho budete potřebovat mít nainstalované prohlížeče, pomocí kterých chcete testovat (nebo jejich headless verze). Zkuste spustit následující skript a v případě problému konzultujte ChatGPT nebo stackoverflow. Pro skript je potřeba mít nainstalovat Chrome. Do URL zadejte nějakou webovou stránku podle vlastního výběru a zkuste na ní vyhledat nějakou HTML značku.

```
from selenium import webdriver
from selenium.webdriver.common.by import By

# Spustí Chrome (musíš mít chromedriver)
driver = webdriver.Chrome()

driver.get("https://example.com")
nadpis = driver.find_element(By.TAG_NAME, "h1")
print(nadpis.text)

driver.quit()
```

7.3. Testování přihlášení

Ukázkový obecný kód pro přihlášení se do webového portálu:

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time

# Spuštění WebDriverů
driver = webdriver.Chrome()

try:
    # Zde zadejte testovanou URL
    driver.get("https://example.com/login")

    # Najdi pole pro jméno a heslo (první parametr je atribut, druhý je hodnota atributu)
    username_input = driver.find_element(By.NAME, "username")
    password_input = driver.find_element(By.NAME, "password")

    # Vyplň údaje
    username_input.send_keys("testuser")
    password_input.send_keys("password123")

    # Odešli formulář
    password_input.send_keys(Keys.RETURN)

    # Počkej na přesměrování
    time.sleep(3)

    # Ověř úspěšné přihlášení - Např. stránka musí obsahovat 'dashboard'
    assert "dashboard" in driver.current_url
    print("Test přihlášení prošel.")

except Exception as e:
    print("Test selhal:", str(e))

finally:
    driver.quit()
```

Úkolem je napsat test, který:

1. Otevře webovou stránku portal.ujep.cz.
2. Najde pole pro uživatelské jméno a heslo.
3. Vyplní přihlašovací údaje.
4. Klikne na tlačítko "Přihlásit se".
5. Ověří, že došlo k úspěšnému přihlášení (např. podle URL nebo textu na stránce).

7.4. Testování navigace

Ukázkový obecný kód pro testování navigace na webovém portálu:

```
driver.get("https://example.com")
about_link = driver.find_element(By.LINK_TEXT, "O nás")
about_link.click()
time.sleep(2)
```

```
assert "about" in driver.current_url
print("Test navigace prošel.")
```

Úkolem je napsat test, který:

1. Ověří, že kliknutí na odkaz Prohlížení (bez přihlášení) funguje.

7.5. Testování formulářů

Následující kód testuje výběr prvků ve formuláři:

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import Select
import time

driver = webdriver.Chrome()
driver.get("https://example.com/products") # nahrad' vlastní URL

try:
    # Najdi a vyber kategorii
    category_select = Select(driver.find_element(By.ID, "category"))
    category_select.select_by_visible_text("Knihy")

    # Najdi a vyber cenové rozpětí
    price_select = Select(driver.find_element(By.ID, "price"))
    price_select.select_by_value("mid") # nebo použij .select_by_index(index)

    # Odešli filtr
    driver.find_element(By.ID, "filterBtn").click()

    # Počkej na načtení výsledků (případně použij WebDriverWait)
    time.sleep(2)

    # Ověř, že výsledky odpovídají filtru
    results = driver.find_element(By.ID, "results").text
    assert "Knihy" in results, "Ve výsledcích chybí kategorie Knihy"
    assert "500 - 2000 Kč" in results or "Cena" in results # příklad

    print("Filtr funguje správně.")

except Exception as e:
    print("Chyba během testu filtru:", str(e))

finally:
    driver.quit()
```

Na následující stránce: ZDE klikněte na tlačítko Podat přihlášku a automatizovaně otestujte vyhledávací formulář vlevo:

1. Zvolte fakultu: FSE - Fakulta sociálně ekonomická
2. Zvolte formu: Kombinovaná
3. Zvolte typ: Bakalářský
4. Zvolte místo studia: Ústí nad Labem
5. Zvolte jazyk: Čeština
6. Klikněte na tlačítko Hledat (pro jistotu)
7. Ověřte, že v pravém seznamu oboru nachází obory: Ekonomika a management, Regionální rozvoj a veřejná správa, Sociální politika a sociální práce.

7.6. Záznam chyby snímkem

Následující kód provede snímek obrazovky v případech kdy test selže.

```
try:
    driver.get("https://example.com/error")
    assert "Neexistující stránka" in driver.page_source
except Exception:
    driver.save_screenshot("error.png")
    print("Screenshot uložen jako error.png")
```

Napište test, který selže na předchozí úlohu a prohlédněte si snímek obrazovky:

1. Zvolte fakultu: FSE - Fakulta sociálně ekonomická
2. Zvolte formu: Kombinovaná
3. Zvolte typ: Bakalářský
4. Zvolte místo studia: Ústí nad Labem
5. Zvolte jazyk: Angličtina
6. Klikněte na tlačítko Hledat (pro jistotu)
7. Ověřte, že v pravém seznamu oboru nachází obor: Ekonomika a management (tento test by měl selhat).
8. Prohlédněte si snímek obrazovky.

7.7. Parametrizované testování

Následující kód ukazuje práci se vstupními daty do formuláře (loginy, hesla) z CSV souboru pro parametrický test.

```
import csv
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time

driver = webdriver.Chrome()

with open("test_users.csv", newline="") as csvfile:
    reader = csv.reader(csvfile)
    for row in reader:
        username, password = row
        driver.get("https://example.com/login")
        driver.find_element(By.NAME, "username").send_keys(username)
        driver.find_element(By.NAME, "password").send_keys(password)
        driver.find_element(By.NAME, "password").send_keys(Keys.RETURN)
        time.sleep(2)

        if "dashboard" in driver.current_url:
            print(f"Uživatel {username}: Přihlášení OK")
        else:
            print(f"Uživatel {username}: Přihlášení SELHALO")

driver.quit()
```

7.8. Testování výkonu

Následující kód měří dobu načítání webové stránky.

```
import time

start_time = time.time()
driver.get("https://example.com")
load_time = time.time() - start_time

print(f"Stránka se načetla za {load_time:.2f} sekund.")
```

Připravte si test výkonu a otestujte, že stránka se načítá do Vámi vyžadované doby.

7.9. Testování responzivity a přístupnosti.

Některé stránky pomocí media-queries v kaskádových stylech nechávají skrývat HTML značky v případě menších rozlišení. Ačkoliv pro to teď nemáme využití na webových stránkách UJEPu, tak si můžeme ukázat, jak se pracuje se zadaným rozměrem displeje přes Selenium:

```
driver.set_window_size(375, 812) # iPhone X rozměry
driver.get("https://example.com")
print("Test probíhá v mobilním zobrazení.")
```

Aria-popisky jsou atributy prvků, kde hodnota představuje nápovědu pro asistivní technologie (čtečky pro slabozraké apod.) [ODKAZ](#). Můžeme napsat jednoduchý test, který pro naší zvědavost ověřuje, že webové stránky vůbec nějaké aria-labels mají:

```
elements = driver.find_elements(By.XPATH, "//*[@aria-label]")
print(f"Počet ARIA prvků na stránce: {len(elements)}")
```

Otestujte, zda přihlášky ke studiu, STAG UJEPu a webové stránky UJEP mají značky pro lepší přístupnost.

7.10. Testování dynamických prvků

Následující kód ukazuje práci s dynamickými prvky. Nenašel jsem na našich stránkách využití, tak jen pro úplnost. Kód testujte, že se tlačítko zobrazí až po 5 sekundách.

```
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

driver.get("https://example.com/dynamic-button")

# Počkej, než se tlačítko objeví
button = WebDriverWait(driver, 10).until(
    EC.presence_of_element_located((By.ID, "delayedButton"))
)

button.click()
print("Test dynamického prvku prošel.")
```

Domácí cvičení - Integrace s Pytest

Pytest může spouštět Selenium testy:

```
import pytest
from selenium import webdriver

@pytest.fixture
def driver():
    driver = webdriver.Chrome()
    yield driver
    driver.quit()

def test_title(driver):
    driver.get("https://example.com")
    assert driver.title == "Expected Title"
```

Přepišť alespoň 5 testů z tohoto cvičení do Pytestu.