

Inżynieria Oprogramowania

Projekt:

„Myśli studenta”

Dorota Dyczek

Nina-Olena Gutovska

Izabela Golec

Natalia Wykpis

Miłosz Wielgus

Vladyslav Khabenets

Testy jednostkowe

Przedstawienie

Testy jednostkowe są częścią procesu testowania oprogramowania, które mają na celu sprawdzenie, czy poszczególne komponenty aplikacji działają poprawnie. W przypadku aplikacji webowych, testy jednostkowe sprawdzają, czy strony są poprawnie wczytywane i czy zawierają oczekiwane elementy.

W poniższym przykładzie przedstawiono zestaw testów jednostkowych dla aplikacji webowej. Testy są napisane przy użyciu biblioteki pytest, która dostarcza narzędzia do tworzenia testów w języku Python.

Środowisko testowe

Aby uruchomić testy jednostkowe, należy mieć zainstalowaną bibliotekę pytest. Można ją zainstalować przy użyciu menedżera pakietów pip: `pip install pytest`

Przygotowanie aplikacji

Przed uruchomieniem testów jednostkowych, należy przygotować aplikację webową. W przykładzie użyto funkcji `create_app()` do utworzenia aplikacji. Dodatkowo, w celu obsługi błędów, ustawiono flagę `PROPAGATE_EXCEPTIONS` na `True` w konfiguracji aplikacji.

Przygotowanie klienta testowego

Aby testować aplikację webową, potrzebujemy klienta testowego, który będzie symulować żądania HTTP do naszej aplikacji. W przykładzie użyto funkcji `test_client()` dostępnej w obiekcie aplikacji Flask, aby utworzyć klienta testowego.

Sprawdzanie wczytywania strony

Test `check_page_loading()` sprawdza, czy strona jest poprawnie wczytywana poprzez wykonanie żądania GET i sprawdzenie, czy status odpowiedzi jest równy 200 (czyli sukces).

Sprawdzanie obecności tagów

Test `check_page_includes_tags()` sprawdza, czy strona zawiera określone tagi HTML. Test ten wykonuje żądanie GET do strony, a następnie sprawdza, czy dany tag jest obecny w treści odpowiedzi.

Testy systemowe

Skrypt instalacji Google Chrome i Chromedriver

Skrypt bash w pliku *install_selenium.sh* służy do automatycznej instalacji przeglądarki Google Chrome i narzędzia Chromedriver w systemie Linux. Skrypt wykonuje następujące czynności:

Aktualizuje repozytorium i wszystkie pakiety systemowe przy użyciu polecenia `apt update` i `apt upgrade`.

Instaluje zależne pakiety, takie jak `wget`, `curl` i `unzip`, przy użyciu polecenia `apt install`.

Pobiera najnowszą wersję pliku `.deb` dla Google Chrome za pomocą polecenia `wget`.

Instaluje Google Chrome przy użyciu polecenia `dpkg -i` na pobranym pliku `.deb`.

Naprawia zależności przy użyciu polecenia `apt --fix-broken install`.

Sprawdza wersję zainstalowanego Google Chrome i pobiera najnowszą dostępną wersję Chromedrivera za pomocą polecenia `curl`.

Porównuje wersję Google Chrome i Chromedrivera. Jeśli są zgodne, kontynuuje instalację.

Pobiera najnowszą wersję Chromedrivera za pomocą polecenia `curl`.

Rozpakowuje pobrany plik `zip` i nadaje mu odpowiednie uprawnienia przy użyciu polecenia `unzip` i `chmod`.

Instaluje bibliotekę Selenium przy użyciu polecenia `pip`.

Zamyka powrót do katalogu domowego za pomocą polecenia `popd`.

Ważne jest, aby przed uruchomieniem skryptu mieć uprawnienia do wykonywania skryptów bash (`chmod +x script.sh`) i mieć zainstalowane pakiety `wget`, `curl`, `unzip` oraz `python3-pip` w systemie.

Skrypt można uruchomić za pomocą polecenia `./script.sh` w terminalu.

Należy pamiętać, że skrypt zakłada, że system jest oparty na dystrybucji Linux i korzysta z menedżera pakietów `apt` do zarządzania oprogramowaniem. Jeśli korzystasz z innego systemu operacyjnego lub menedżera pakietów, skrypt może nie działać poprawnie i wymagać dostosowania.

Test 1: test_empty_list_courses_button

Test sprawdza, czy po kliknięciu przycisku na stronie internetowej, lista kursów jest pusta.

Kroki testowe:

Otwórz stronę internetową.

Znajdź element przycisku na stronie o podanym identyfikatorze.

Kliknij przycisk.

Znajdź element <div> o identyfikatorze 'course_list' i pobierz jego zawartość HTML.

Porównaj pobraną zawartość HTML z oczekiwanym pustym ciągiem znaków.

est 2: test_list_courses_button

Test sprawdza, czy po wybraniu opcji 'Bydgoszcz' i kliknięciu przycisku na stronie internetowej, lista kursów jest poprawnie wyświetlana.

Kroki testowe:

Otwórz stronę internetową.

Znajdź element opcji odpowiadający wartości 'Bydgoszcz' i kliknij go.

Kliknij przycisk.

Znajdź element <div> o identyfikatorze 'course_list' i pobierz jego tekst.

Porównaj pobrany tekst z oczekiwaną zawartością listy kursów.