# ORW Ćwiczenie 7 – Testy aplikacji WWW

Autor: Kornel Ostrowski

Grupa: 125NCI\_B

Link do projektu: <https://github.com/ostrowskik/TestWWW_orw_cw7>

**Zadania do zrealizowania:**

1. Za pomocą wybranego przez siebie narzędzia zademonstruj test wybranej strony WWW.
2. Bądź gotowy do prezentacji możliwości i modyfikacji stworzonych testów.

**Przykładowe pytania:**

* Na czym polegają testy aplikacji WWW?

Testy aplikacji WWW polegają na przeprowadzeniu wielokrotnego i wielowątkowego sprawdzenia funkcjonalności poszczególnych elementów aplikacji WWW.

* Przedstaw ideę Test Driven Development (TDD)

Ideą TDD jest stworzenie testów sprawdzających funkcjonalność aplikacji zanim zaczniemy tworzyć kod źródłowy programu, pod te testy będzie on przygotowywany.

* Scharakteryzuj główne rodzaje testów w TDD:
  + testy jednostkowe (*unit tests*) — testujemy pojedynczą, jednostkową część kodu: zazwyczaj klasę lub metodę;
  + testy integracyjne (*integration tests*) — testujemy kilka komponentów systemu jednocześnie;
  + testy regresyjne (*regression tests*) — po wprowadzeniu naszej zmiany uruchamiane są wszystkie testy w danej domenie biznesowej celem sprawdzenia czy zmiana nie spowodowała błędu w innej części systemu;
  + testy akceptacyjne (*acceptance tests*) — testy mające na celu odpowiedzieć na pytanie czy aplikacja spełnia wymagania biznesowe.
* Scharakteryzuj cykl Red-Green-Refactor w TDD:

**Red**: Piszemy test, który się nie powodzi.

Testy piszemy do pustych, ale istniejących już klas i metod (dzięki czemu możemy korzystać z IntelliSense).

Uruchamiamy test i oczekujemy, że się nie powiedzie.

**Green**: Piszemy kod aby testy się powiodły.

Implementujemy kod (według dokumentacji).

Uruchamiamy testy. Wszystkie testy muszą się powieść.

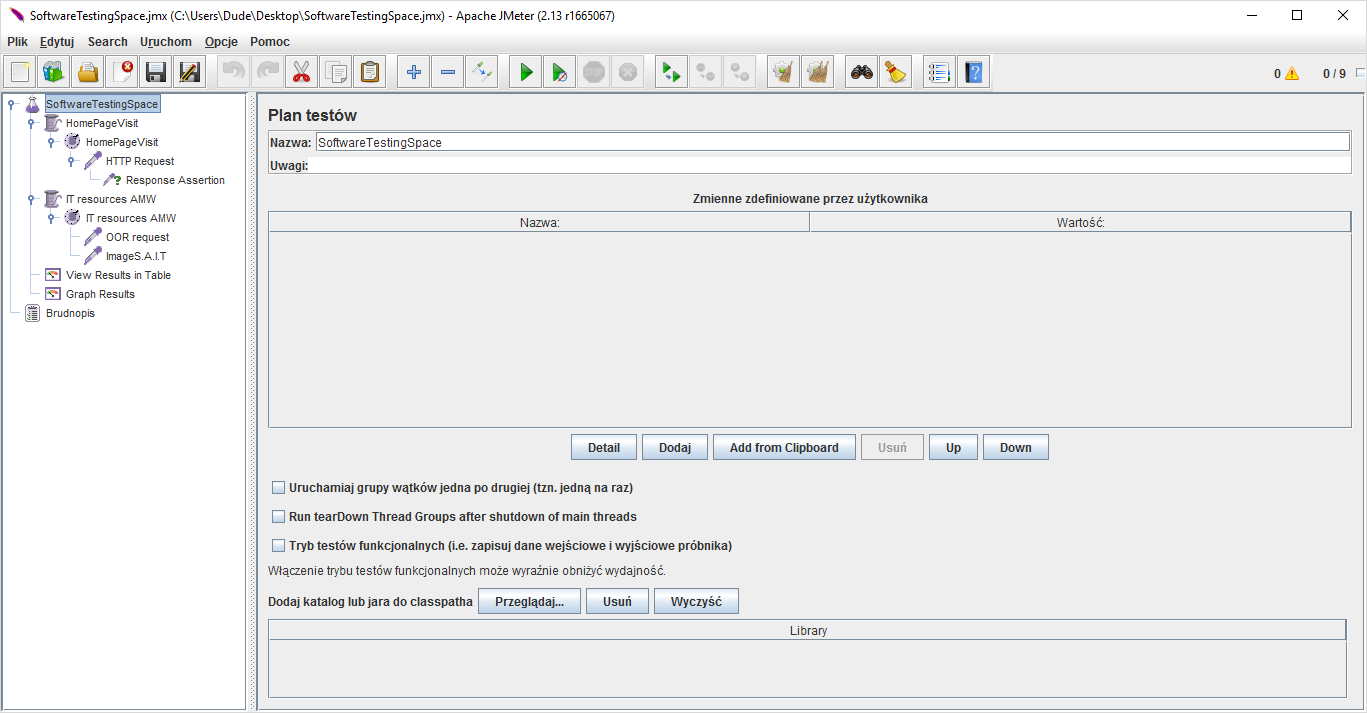
**Refactor**: Refaktoryzacja kod—wprowadzenie zmian, które poprawiają jakość kodu (np. usunięcie duplikacji), ale nie zmieniają jego funkcjonalności.

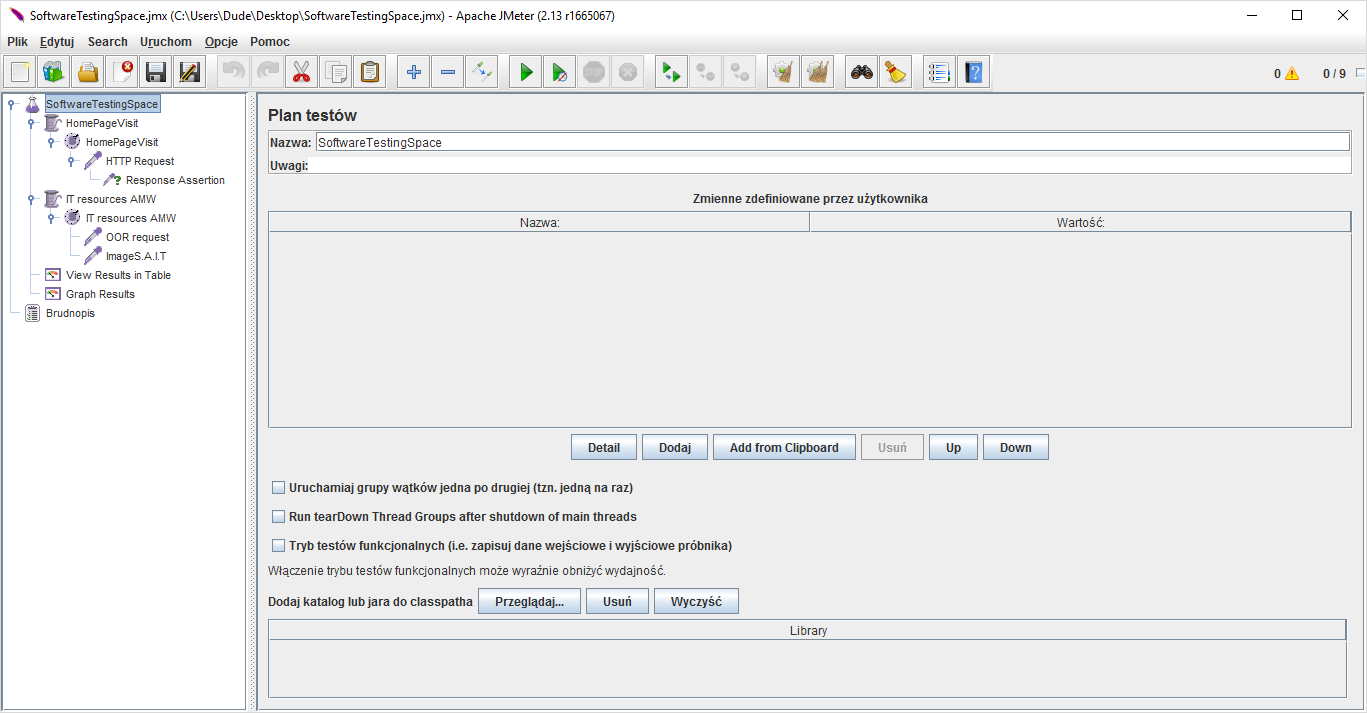
Po refaktoryzacji, uruchamiamy wszystkie testy by sprawdzić czy czegoś nie zepsuliśmy.

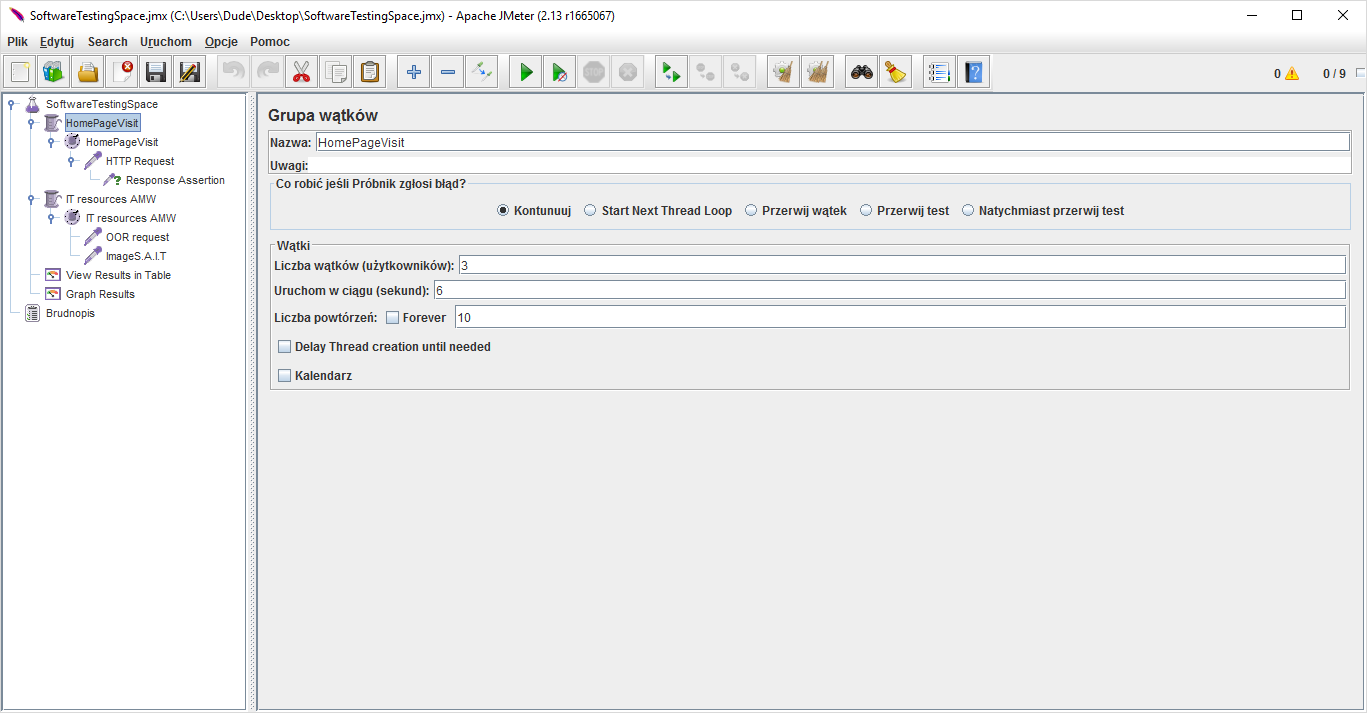
Ten punkt jest często lekceważony lub pomijany w procesie. Nie zapominajmy o tym, równie ważnym co dwa poprzednie, elemencie.

# Realizacja zadań

Test został przeprowadzony w aplikacji Apache JMeter:

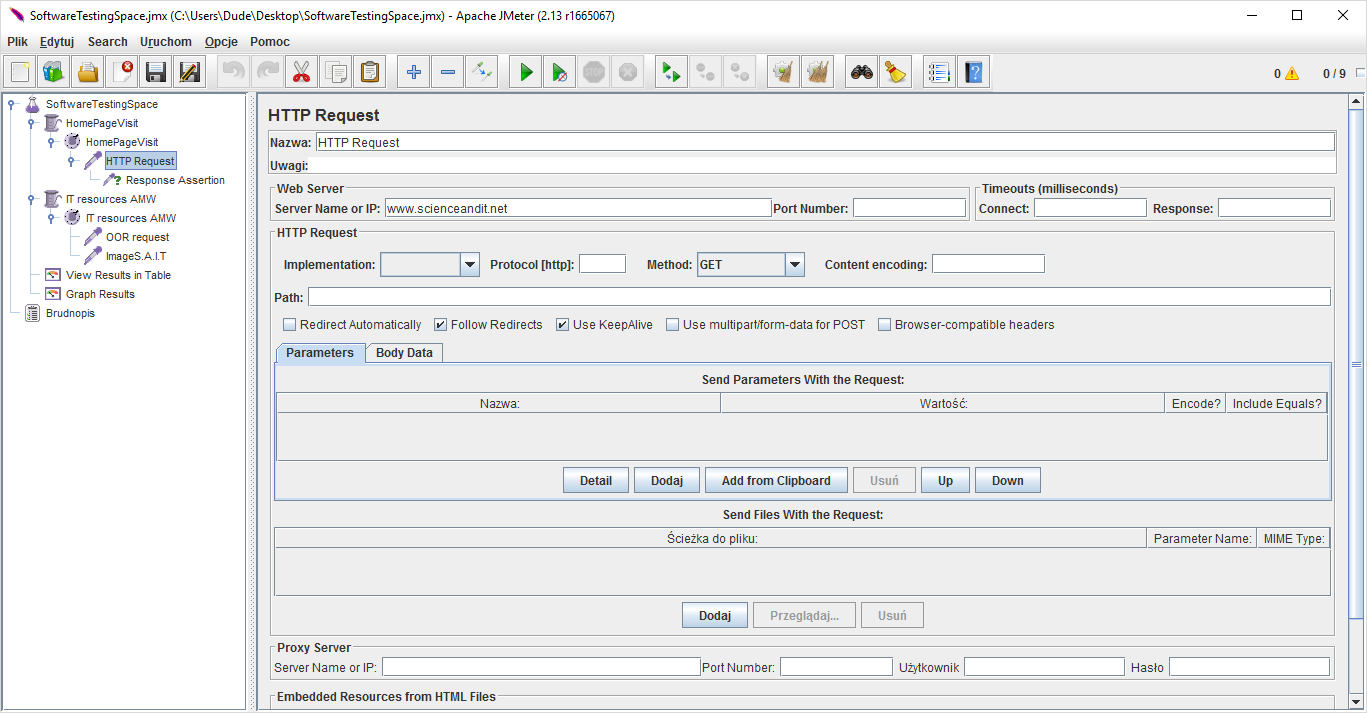


  
Zdefiniowane zostały 2 grupy wątków: ‘HomePageVisit’ oraz ‘IT resources AMW’

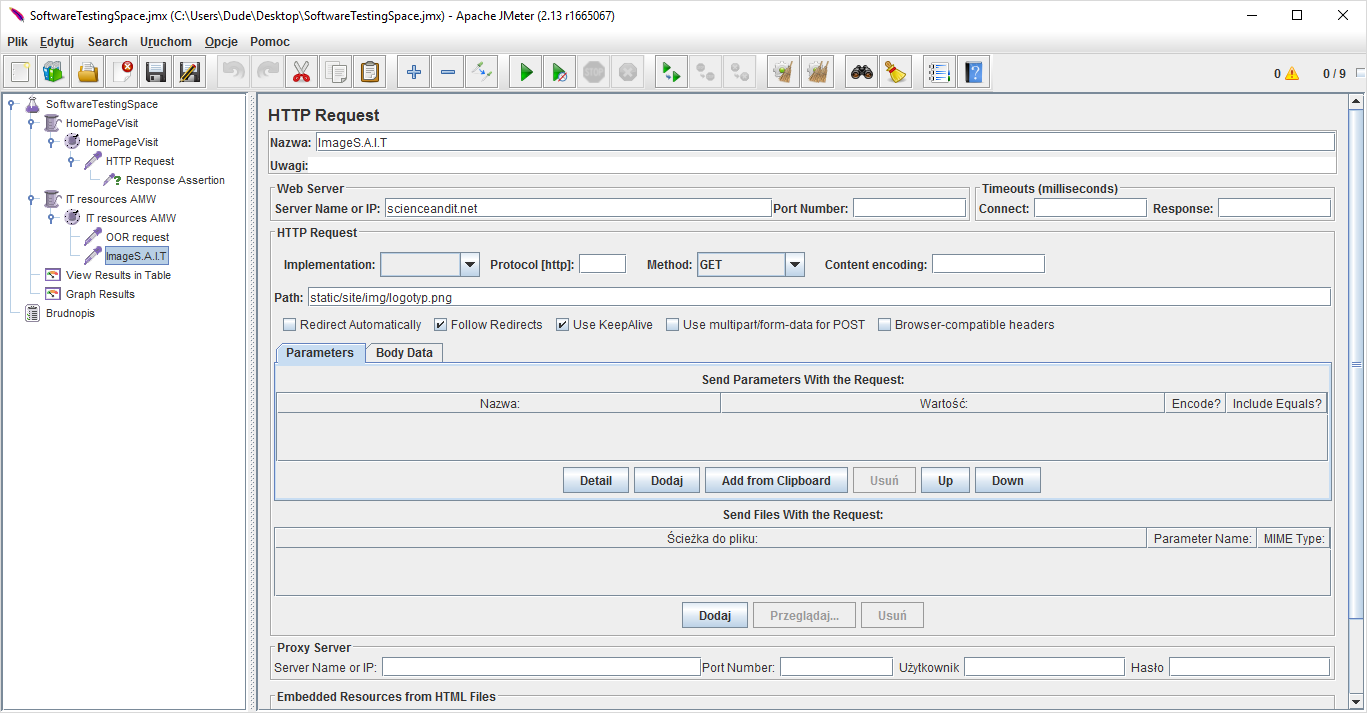


Tutaj określamy liczbę wątków(użytkowników), czas w jakim ma być to wykonane oraz ilość powtórzeń danego zapytania (wywołania).

Kolejno do każdej z grup z wątków dodany został prosty kontroler testu, pod którego zostały podpięte Samplery (tzw. próbniki, modele), które określają nam sposób sprawdzenia zasobu.



Mamy tutaj wywołanie strony domowej metodą HTTP GET lub w drugim przypadku odniesienie się do obrazka zawartego na stronie.



Dane z testu można zebrać w postaci tabeli lub wykresu:

