

PLAN TESTÓW

API <https://reqres.in/api/use>

Wersja 1.0

HISTORIA ZMIAN

Wersja	Data	Autor	Komentarz
1.0	2.11.2023	Paweł Aksman	Utworzenie dokumentu

METRYKA

Nazwa dokumentu	Plan testów API https://reqres.in/api
Autor	Paweł Aksman
Opracowany na potrzeby	XYZ Company
ID	RR-021123
Wersja	1.0
Data	5.03.2024
Liczba stron	20

1. Cel dokumentu

Celem niniejszego dokumentu jest dostarczenie informacji dotyczących organizacji, przeprowadzenia testów oraz dokumentacji testów. W dalszej części dokumentu zawarte zostaną informacje dotyczące:

- przedmiotu testów
- opisu przedmiotu testów
- elementów testowych
- zakresu testów
- technik projektowania testów
- wymagań dotyczących danych testowych danych testowych
- środowiska testowego
- wymagań dotyczących środowiska testowego
- kryteriów wejścia/wyjścia
- kryteriów zaliczenia testów
- kryteriów zakończenia testów
- listy narzędzi, które zostaną wykorzystane w procesie testowania
- realizacji testów
- raportowania testów

Dokument został opracowany przez Wykonawcę oraz dostarczony do zamawiającego w celu akceptacji tego dokumentu.

2. Przedmiot testów

Przedmiotem testów jest weryfikacja funkcjonalności oraz przeprowadzenie testów wydajnościowych interfejsu API <https://reqres.in/api>.

3. Opis przedmiotu testów

<https://reqres.in/api> jest źródłem udostępniającym publiczny interfejs API. Jest to interfejs stworzony z myślą o użytkownikach chcących zgłębić tematykę testowania API za pomocą przeznaczonych do tego narzędzi. Umożliwia on operacje oparte na poleceniach **GET**, **POST**, **PUT**, **PATCH**, **DELETE**. Dostęp do interfejsu jest bezpłatny oraz nieograniczony.

4. Elementy testowe

Testom funkcjonalnym oraz нефункциональным тестом wydajnościowym podlegać będą:

- pobieranie listy użytkowników
- pobieranie danych pojedynczego użytkownika
- pobieranie listy zasobów
- pobieranie danych pojedynczego zasobu
- tworzenie nowego użytkownika
- aktualizacja danych użytkownika
- usuwanie użytkownika
- autoryzacja
- logowanie
- pobieranie zasobów z opóźnioną reakcją

5. Zakres testów

Element testowy	Zakres testowania
Pobieranie listy użytkowników	Testy funkcjonalne
	Testy wydajnościowe
Pobieranie danych pojedynczego użytkownika	Testy funkcjonalne
	Testy wydajnościowe
Pobieranie zasobów	Testy funkcjonalne
	Testy wydajnościowe
Pobieranie danych pojedynczego zasobu	Testy funkcjonalne
	Testy wydajnościowe
Tworzenie nowego użytkownika	Testy funkcjonalne
	Testy wydajnościowe
Aktualizacja danych użytkowników	Testy funkcjonalne
	Testy wydajnościowe
Usuwanie użytkownika	Testy funkcjonalne
	Testy wydajnościowe
Autoryzacja	Testy funkcjonalne
	Testy wydajnościowe
Logowanie	Testy funkcjonalne
Pobieranie zasobów z opóźnioną reakcją	Testy funkcjonalne
	Testy wydajnościowe

6. Techniki projektowania testów

W testowaniu czarnoskrzynkowym wykorzystane zostają następujące techniki:

- przypadki użycia
- klasy równoważności
- tablice decyzyjne

7. Wymagania dotyczące danych testowych

- dla testów pozytywnych należy przygotować pliki o formacie JSON zawierające wszystkie klucze wymagane przez aplikację
- dla testów pozytywnych testowych należy przygotować pliki JSON, które mogą zawierać opcjonalne klucze, zgodnie z wymaganiami aplikacji
- dla testów negatywnych należy przygotować dane testowe zawierające nieprawidłowe nazwy kluczy, z punktu widzenia aplikacji
- dla testów negatywnych należy przygotować dane testowe, które nie zawierają wymaganych przez aplikację kluczy
- dla testów negatywnych należy przygotować dane zawierające pliki JSON w niepoprawnym formacie
- dla testów zarówno pozytywnych, jak i negatywnych, zaleca się przygotowanie danych zawierających schematy JSON w celu porównania ich z treścią odpowiedzi

8. Środowisko testowe

Nazwa komputera	pawelaksman-desktop
System operacyjny	Ubuntu 22.04.4
Język	Polski
Producent systemu	To Be Filled by O.E.M.
Model systemu	To Be Filled by O.E.M.
BIOS	P1.90
Procesor	Intel(R) Core(TM) i5-4440 CPU @ 3.10GHz
Pamięć	160114988 kB
Dostępna pamięć systemu operacyjnego	12GB
Przestrzeń wymiany	<ul style="list-style-type: none">• type: file• size: 2G• prio: -2
Wersja OpenGL	4.6 (Compatibility Profile) Mesa 23.2.1-1ubuntu3.1~22.04.2
Wersja DxDiag	10.00.19041.2075 64bit Unicode

Ponadto zapewnione zostanie łącze internetowe o szybkości 100.0 Mb/s.

9. Wymagania dotyczące środowiska testowego

- system operacyjny powinien być na bieżąco sprawdzany pod kątem dostępnych aktualizacji
- dla przeprowadzenia testów automatycznych należy zapewnić najnowszą wersję IDE, w których zostaną zaimplementowane skrypty testowe
- należy na bieżąco sprawdzać dostępność najnowszych wersji dla narzędzi, w których zostaną zaimplementowane skrypty testowe
- należy na bieżąco sprawdzać dostępność najnowszych wersji dla narzędzi przeznaczonych do przeprowadzenia testów wydajnościowych

10. Kryteria wejścia/wyjścia

Za kryteria wejścia uznaje się:

- poprawnie skonfigurowane zgodnie z wymaganiami oraz gotowe do przeprowadzenia testów środowisko testowe.
- skrypty testowe zaimplementowane zgodnie z panującymi dobrymi praktykami pisania kodu
- funkcjonalność, która podlega testowaniu ma zakończoną fazę implementacji
- przygotowane oraz dostępne dane testowe

Za kryteria wyjścia uznaje się:

- rezultaty przeprowadzonych testów pokrywają się z wymaganiami zawartymi w dokumentacji oraz historyjkach użytkownika

11. Kryteria zaliczenia testów

- dla testów pozytywnych kod odpowiedzi jest zgodny ze specyfikacją:
 - **200** dla metody **GET**
 - **201** dla metod **POST** lub **PUT** tworzących nowy zasób
 - **200, 202, 204** dla metody **DELETE**
- dla testów pozytywnych treść odpowiedzi jest dobrze sformułowanym obiektem JSON
- dla testów pozytywnych struktura odpowiedzi jest zgodna z modelem danych (weryfikacja schematu: nazwy pól i typy pól są zgodne z oczekiwaniami, łącznie z obiektami zagnieżdżonymi; wartości pól są zgodne z oczekiwaniami; pola nie dopuszczające wartości null nie posiadają wartości null)
- dla testów pozytywnych nagłówki są zgodne ze specyfikacją
- dla testów negatywnych kod odpowiedzi nie wynosi 2XX
- dla testów negatywnych kod odpowiedzi jest zgodny z przypadkiem błędu zdefiniowanym w specyfikacji
- dla testów negatywnych format błędu jest zgodny ze specyfikacją- jest on prawidłowym obiektem JSON lub obiektem XML
- dla testów negatywnych sprawdź błędu jest poprawny dla danego przypadku błędu i zgodny ze specyfikacją
- dla testów wydajnościowych czas odpowiedzi serwera dla pojedynczych żądań jest nie większy niż 1500ms dla 10 użytkowników
- dla testów wydajnościowych czas odpowiedzi serwera dla pojedynczych żądań jest nie większy niż 1500ms dla 30 użytkowników
- dla testów wydajnościowych średni czas odpowiedzi serwera dla wszystkich żądań jest nie większy niż 1500ms dla 30 użytkowników

12. Kryteria zakończenia testów

Za kryteria zakończenia uznano czas, w którym zakończono przeprowadzanie testów.

13. Lista narzędzi, które zostaną wykorzystane w celu przeprowadzenia testów

Podczas przeprowadzania testów wykorzystane będą następujące narzędzia:

WebAIM	Testy użyteczności
Apache JMeter	Testy wydajnościowe
Microsoft Visual Studio Code	Przygotowanie danych testowych
IntelliJ Idea	Automatyzacja testów
REST Assured	Automatyzacja testów
TestNG	Automatyzacja testów
Allure	Raportowanie
Qase	Zarządzanie testami

Języki programowania wykorzystane podczas automatyzacji testów

REST Assured	Java
---------------------	------

14.Realizacja testów

Kolejność wykonywania testów jest zgodna z kolejnością przypadków testowych w poszczególnych scenariuszach. W przypadku uzyskania negatywnego wyniku testu, dany przypadek testowy zostanie powtórzony. Po ponownym uzyskaniu wyniku negatywnego, tester zawrze informacje opisujące występujący defekt w raporcie z defektów. W raporcie tym zostaną również zamieszczone informacje dotyczące danych testowych (jeżeli są) wykorzystane w celach testów, oraz kroków reprodukcji, jakie należy wykonać w celu powtórzenia testu.

15.Raportowanie testów

Po przeprowadzeniu testów, przedstawiciel Wykonawcy dostarczy Zamawiającemu wyniki testów, na które składać będą się wypełnione w trakcie procesu testowania scenariusze testowe oraz raporty testów automatycznych wygenerowane przez narzędzie Allure oraz raporty testowe wygenerowane przez narzędzie Qase. W przypadku braku możliwości przeprowadzenia testów z przyczyn stojących po stronie Zamawiającego, przedstawiciel Wykonawcy dostarczy raporty z pominięciem scenariuszy nie przeprowadzonych testów.