**І рівень**

**1. Створи набір з GET, POST, PUT, PATCH, DELETE запитів до** [**JSONPlaceholder**](https://jsonplaceholder.typicode.com/)**, які надсилаються протягом 10 секунд у 3 ітерації.**

**2. До кожного з запитів застосуй 3 різних assert’и.**

**3. Результати виконання тестів мають бути отримані за такими Listener’ами:**

* **View Results in Tree**
* **Summary Report.**

**4. Опиши висновки щодо результатів тестування (базуючись на репортерах) в окремому файлі.**

Я тестувала усі 5 методи, обравши assertions (Size and Duration). Мені важливо було, щоб мої запити вміщались у час завантаження, та розмір завантаження.

Для 1 методу Post було вказано, що Size <= 2000 bytes; Duration = 1500 ms.;

Для 2 методу Get було вказано, що Size <= 2000 bytes; Duration = 2000 ms.;

Для 3 методу Patch було вказано, що Size <= 2000 bytes; Duration = 2000 ms.;

Для 4 методу Put було вказано, що Size <= 1500 bytes; Duration = 1500 ms.;

Для 5 методу Delete було вказано, що Size <= 2000 bytes; Duration = 2000 ms.

За допомогою звіту у View Results Tree видно, що вимоги були виконані і код статус у запиті Post - 201, у решти 4 запитів - 200, що означає про успішність проведення навантажувального тестування для кожного запиту.

Також ставилися умови, що запити надсилаються протягом 10 секунд у 3 ітерації, і згідно звіту “Summary Report” у кожному запиті видно, що вони дійсно надсилались тричі, і час надсилання також виконався.

**5. Створений тест-план та документ з описом результатів виконання репортерів додай в свій репозиторій в GitHub.**

[**https://github.com/AnushShaboian/NewRepository\_AnushShaboian.git**](https://github.com/AnushShaboian/NewRepository_AnushShaboian.git)

***Додай посилання на відповідний репозиторій в LMS.***

**II рівень**

**1. Виконай завдання попереднього рівня.**

**2. Створи новий тест-план на основі тест-плану з попереднього рівня. В ньому:**

* **для кожного із запитів зроби стрес-тест використаного API;**

**1. Запит Post:**

**Assertion - Headers size <= 3000 bets.**

**Thread Properties:**

**200/1/2 - without errors;**

**3000/1/2 - without errors;**

**10000/1/2 - errors 83%**

(Response code:Non HTTP response code: java.net.SocketException

Response message:Non HTTP response message: Connection reset).

**2. Запит Get:**

**Assertion - Check Response Code = 200.**

**Thread Properties:**

**200/5/1 - without errors;**

**2700/5/1 - without errors**

**10000/5/1 - errors 87%**

(Response code:Non HTTP response code: java.net.SocketException

Response message:Non HTTP response message: Connection reset);

**3. Запит Patch:**

**Assertion - Check Body Size <= 3000 byts.**

**Thread Properties:**

**200/1/2 - without errors;**

**2000/1/2 - without errors**

**20000/1/2 - errors 92%**

(Response code:Non HTTP response code: java.net.SocketException

Response message:Non HTTP response message: Connection reset);

**4. Запит Put:**

**Assertion - Check format = XML.**

**Thread Properties:**

**200/1/2 - without errors;**

**2200/1/2 - without errors**

**20000/1/2 - errors 98%**

(Response code:Non HTTP response code: java.net.SocketException

Response message:Non HTTP response message: Connection reset);

**5. Запит Delete:**

**Assertion - Duration Assertion = 1000 ms.**

**Thread Properties:**

**200/1/2 - without errors;**

**500/1/2 - without errors**

**2000/1/2 - errors 46,62%**

(Duration time is more than 1000 ms);

* **потрібно виявити такі мінімальні комбінації параметрів, за яких вебсервіс перестає витримувати навантаження.**

**Параметри:**

**Post - 3000/1/2**

**Get - 2700/5/1**

**Patch - 2000/1/2**

**Put - 2200/1/2**

**Delete - 500/1/2**

**3. Створений тест-план додай у свій репозиторій в GitHub.**

[**https://github.com/AnushShaboian/NewRepository\_AnushShaboian.git**](https://github.com/AnushShaboian/NewRepository_AnushShaboian.git)

***Додай посилання на відповідний репозиторій в LMS.***