**1. Створи набір з GET, POST, PUT, PATCH, DELETE запитів до JSONPlaceholder, які надсилаються протягом 10 секунд у 3 ітерації.**

**Дата:** 2025-09-20

**Test Plan:** JSONPlaceholder.jmx

**Thread Group:** 20 user, 3 ітерації, ~10 секунд

**Тестовані endpoints**

- GET /posts/1

- POST /posts

- PUT /posts/1

- PATCH /posts/1

- DELETE /posts/1

**Результати (Summary Report)**

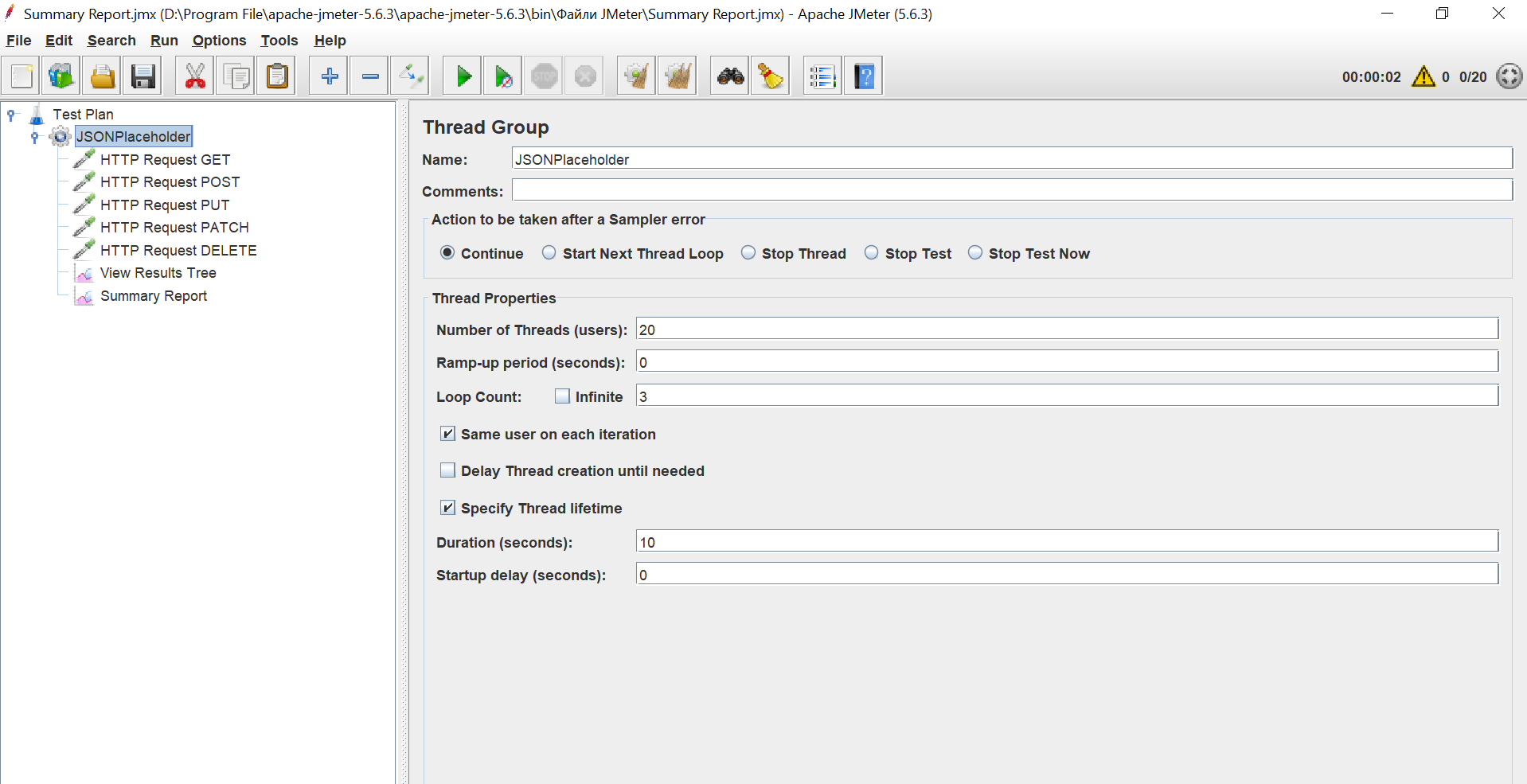
- Всі запити успішні (Response Code 200/201)

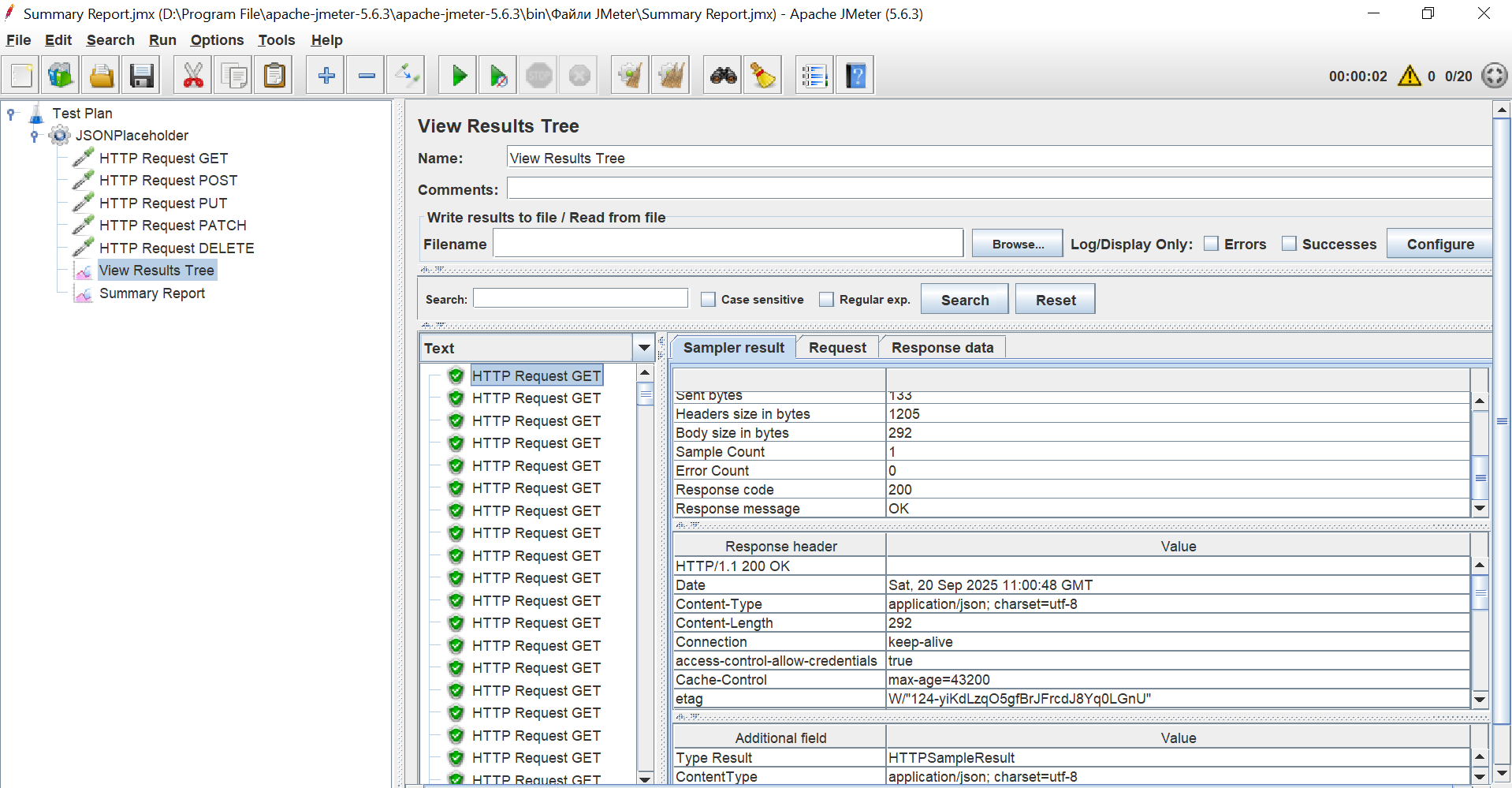
- Середній час відповіді: 159 ms

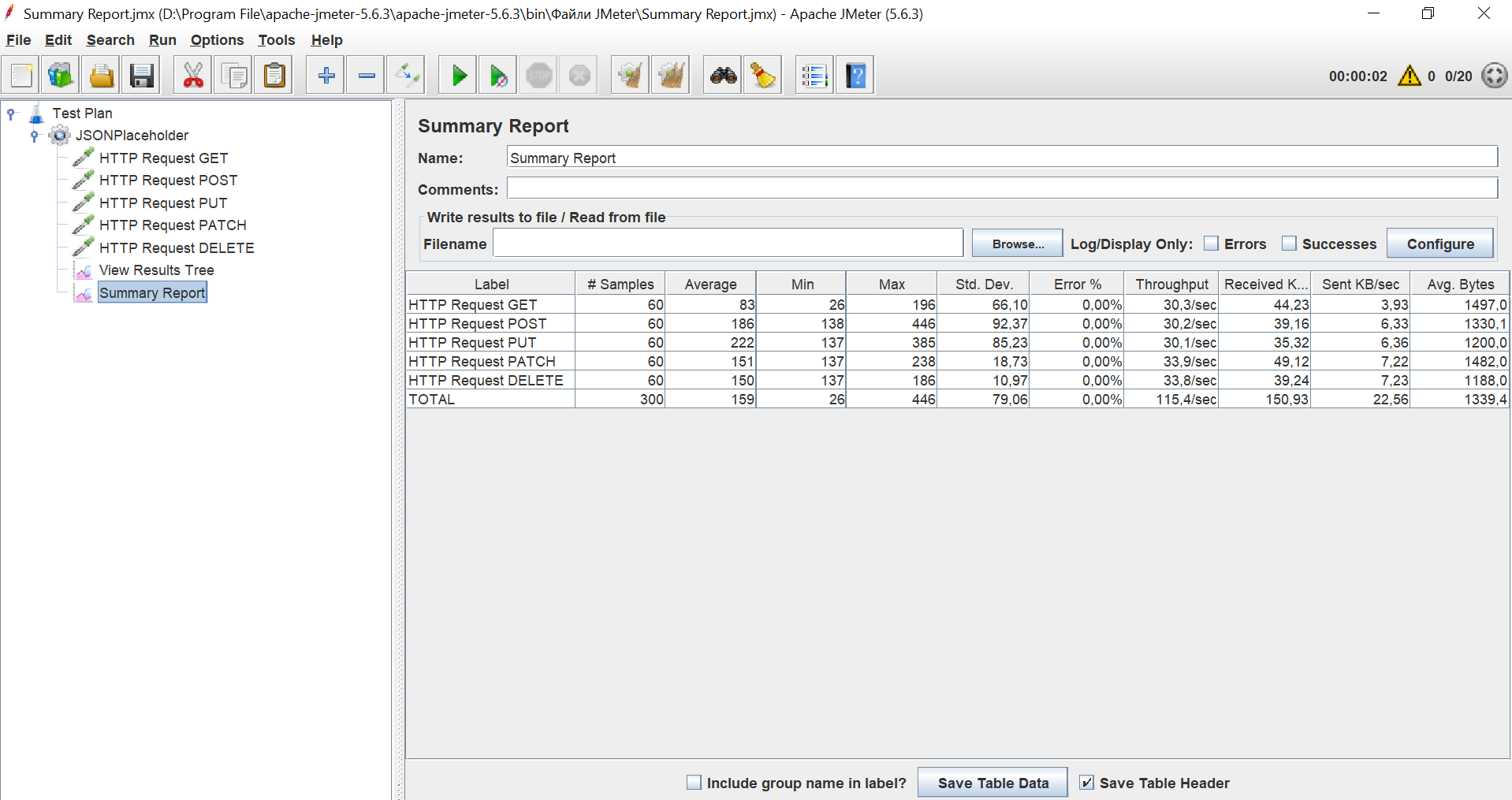
- Error %: 0%

**Висновок**

- Тест виконано успішно, всі запити пройшли без помилок.







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

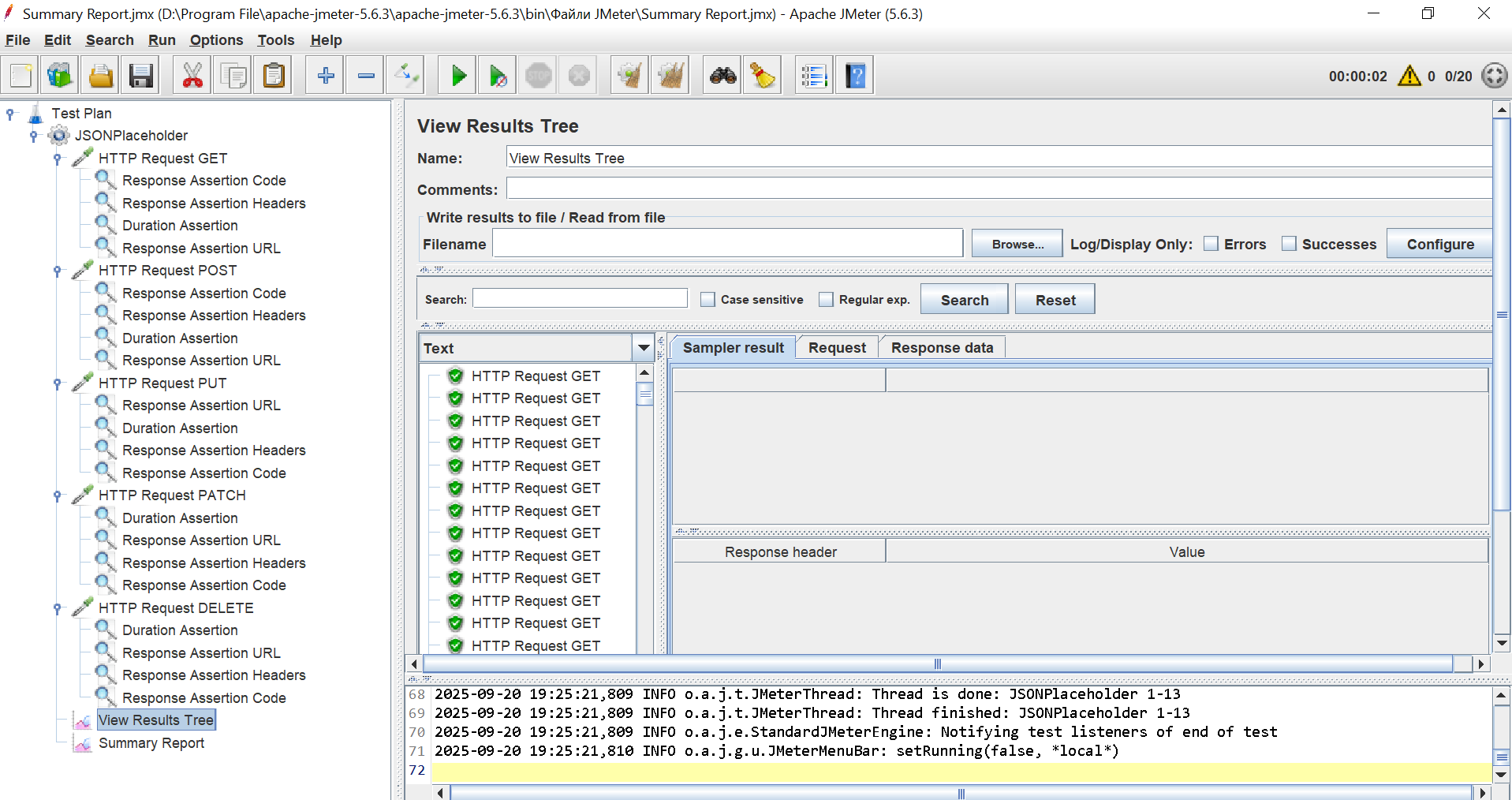
**Assertions**

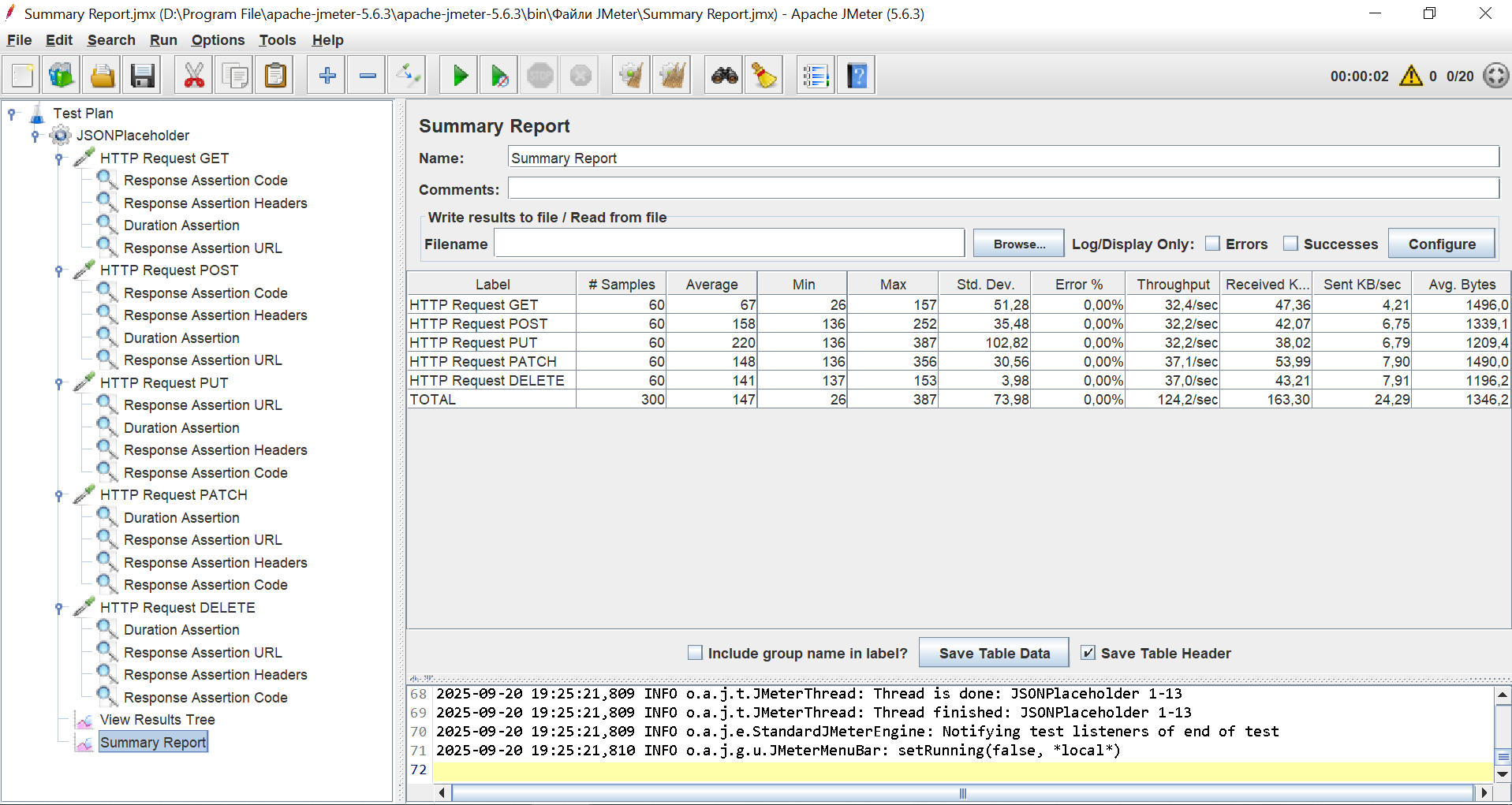
*1. Response Code – 200 / 201*

*2. Response Header - json*

*3. Duration Assertion – 5000mlsc*

*4. Response URL - /posts/1 and /posts*





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

**1. Опис тесту:**

- Виконано HTTP запити: GET /posts/1, POST /posts, PUT /posts/1, PATCH /posts/1, DELETE /posts/1

- Кожен запит тестувався з чотирма Assertions:

1. Response Code – 200 / 201

2. Response Header - json

3. Duration Assertion – 5000mlsc

4. Response URL - /posts/1 and /posts

**2. Використані Listener:**

- View Results Tree – для детальної перевірки кожного запиту

- Summary Report – для зведених показників: час відповіді, кількість успішних/провалених запитів

**3. Результати:**

- Всі HTTP запити пройшли успішно (Response Code відповідає очікуваному)

- Assertions у всіх запитах пройшли, Status – зелений

- Середній час відповіді (Total) для всіх запитів 147 ms

- Жодних помилок у View Results Tree не виявлено

**4. Висновки:**

- Сервер JSONPlaceholder стабільно обробляє запити всіх методів (GET, POST, PUT, PATCH, DELETE)

- Response Assertions та Duration Assertions підтверджують коректність і швидкість обробки запитів

<https://github.com/AndreyMaslo4/GitLesson/tree/main/%D0%A3%D1%80%D0%BE%D0%BA%20%D0%BD%D0%B0%D0%B2%D0%B0%D0%BD%D1%82%D0%B0%D0%B6%D0%B5%D0%BD%D0%BD%D1%8F>

**Створи новий тест-план на основі тест-плану з попереднього рівня.**

**В ньому:**

* **для кожного із запитів зроби стрес-тест використаного API;**
* **потрібно виявити такі мінімальні комбінації параметрів, за яких вебсервіс перестає витримувати навантаження.**

**У Thread Group налаштовано:**

*Кількість користувачів (Number of Threads): від 50 до 500*

*Ramp-Up Period: 10 секунд*

*Loop Count: 1 (один цикл)*

*Duration: 60 секунд*

*До кожного запиту додано 4 асерти:*

*1. Response Code – 200 / 201*

*2. Response Header - json*

*3. Duration Assertion – 5000mlsc*

*4. Response URL - /posts/1 and /posts*

***Використані лістенери:***

*View Results Tree – для перегляду детальних відповідей сервера та відладки.*

*Summary Report – для збору агрегованої статистики (кількість запитів, середній час, відсоток помилок, Throughput).*

***Результати***

*Усі запити (GET, POST, PUT, PATCH, DELETE) виконались успішно при 50 віртуальних користувачах.*

*Середній час відповіді: ~130–600 мс.*

*Максимальний час відповіді: ~1098 мс.*

*Всі асерти спрацювали коректно, Response Code = 200 / 201, Duration < 5000 мс.*

*Відсоток помилок: 0%.*

***Висновки***

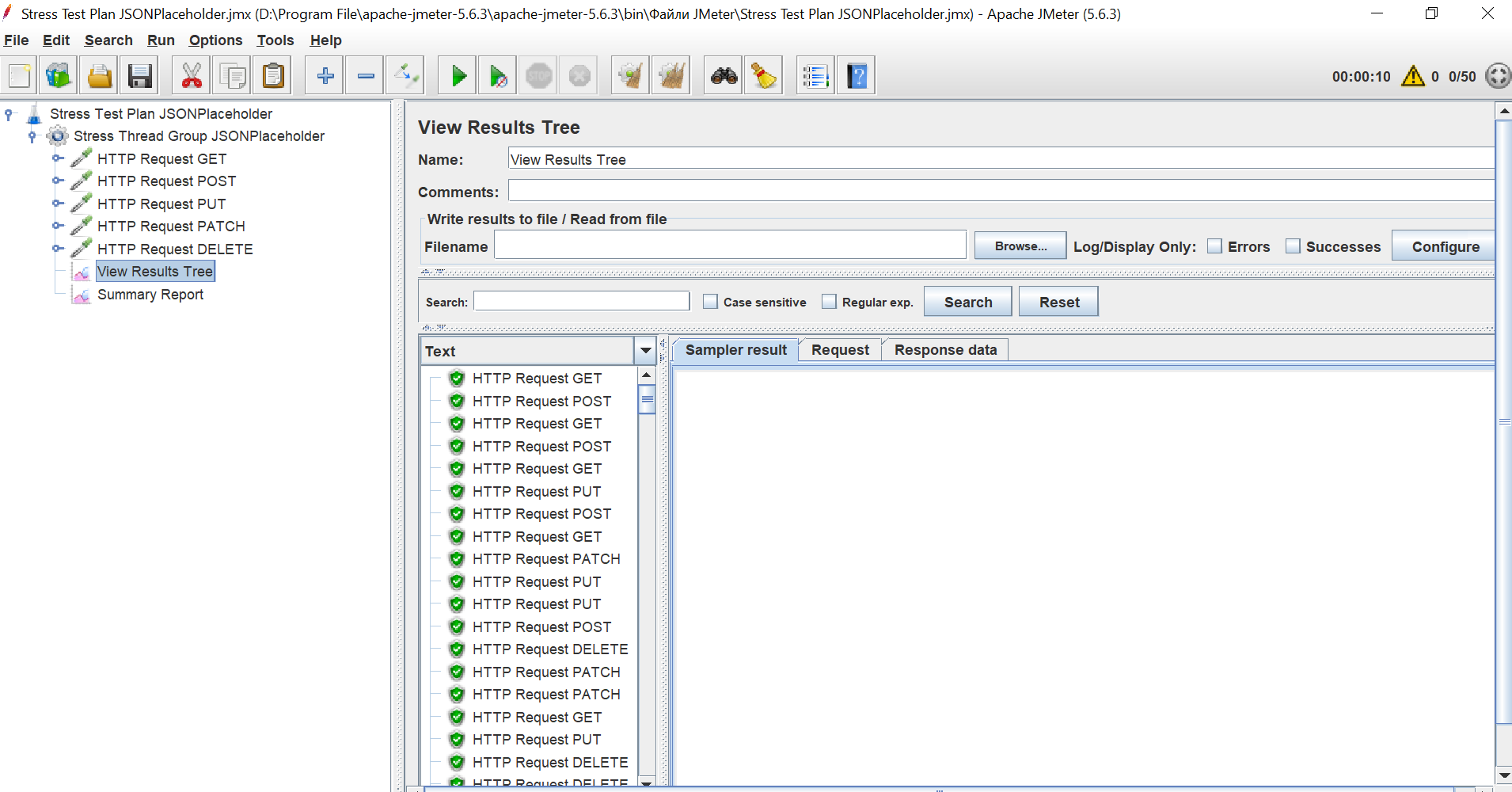
*Тестоване API витримує навантаження у від 50 до 500 одночасних користувачів без втрати стабільності.*

*Всі HTTP-методи (GET, POST, PUT, PATCH, DELETE) працюють коректно та повертають очікувані коди відповіді.*

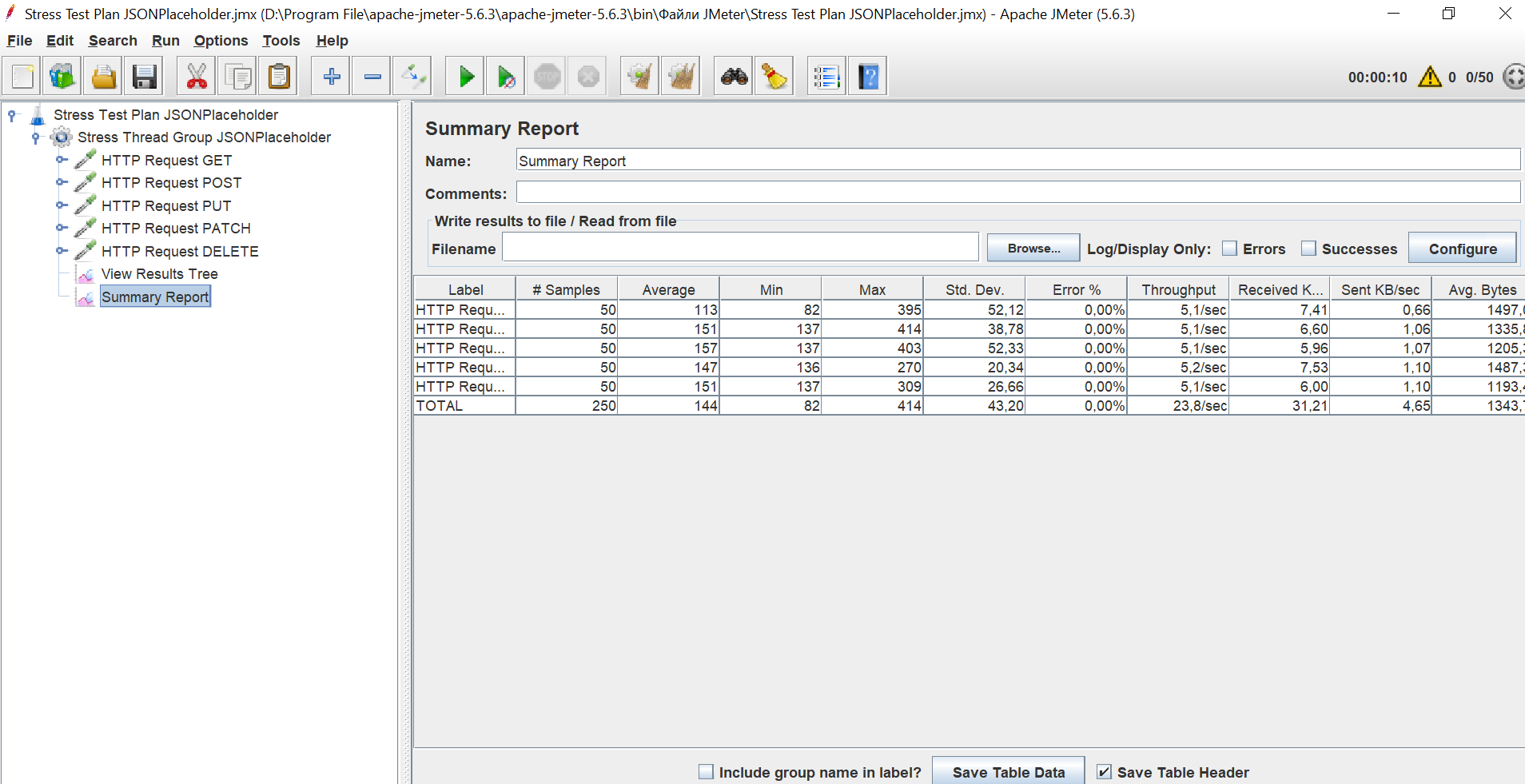
*Сервіс зберігає працездатність при стрес-навантаженні, час відгуку вкладається у встановлені 5 секунд.*

**Кількість користувачів = 50**

*View Results Tree*

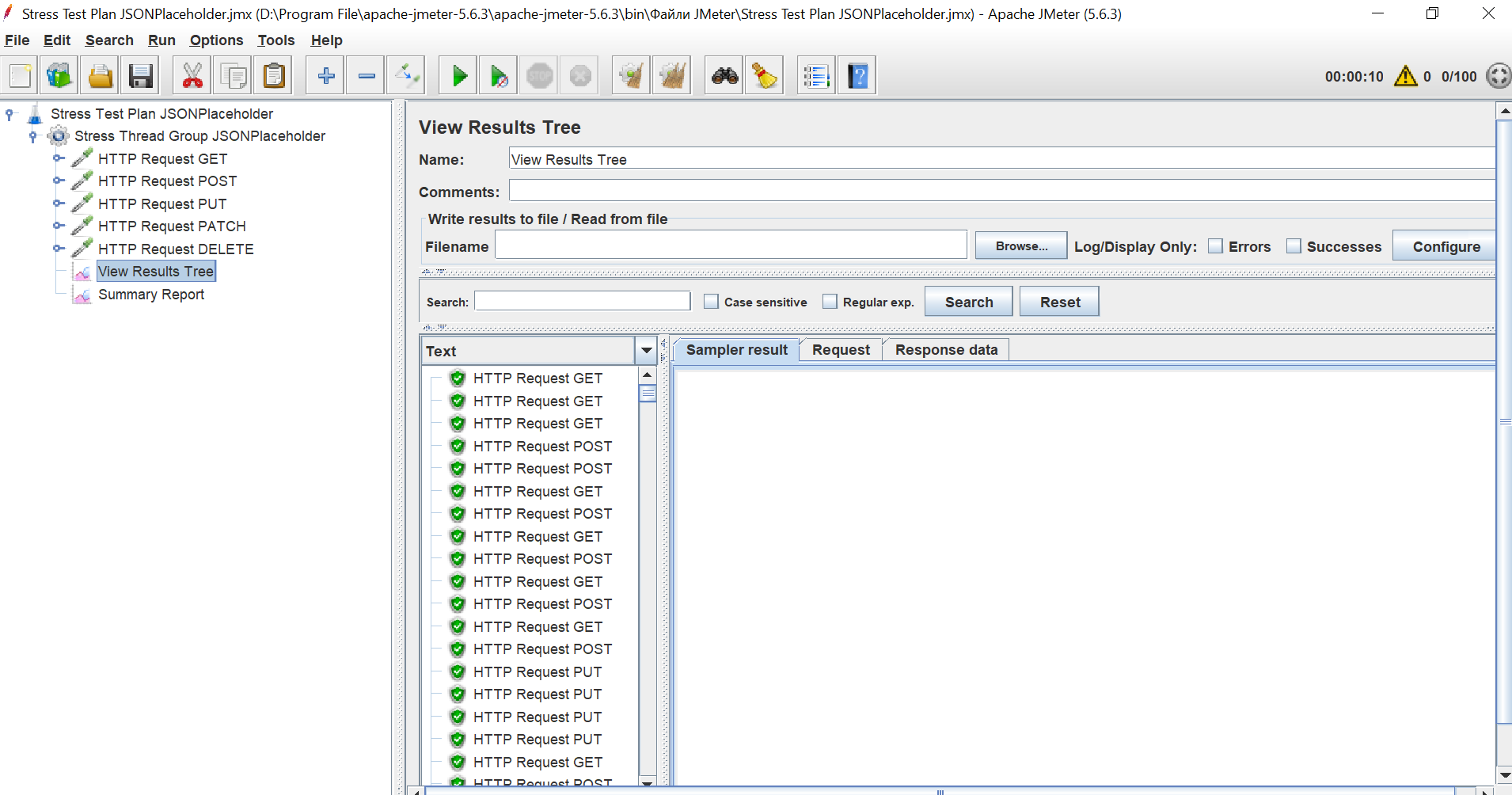


*Summary Report*

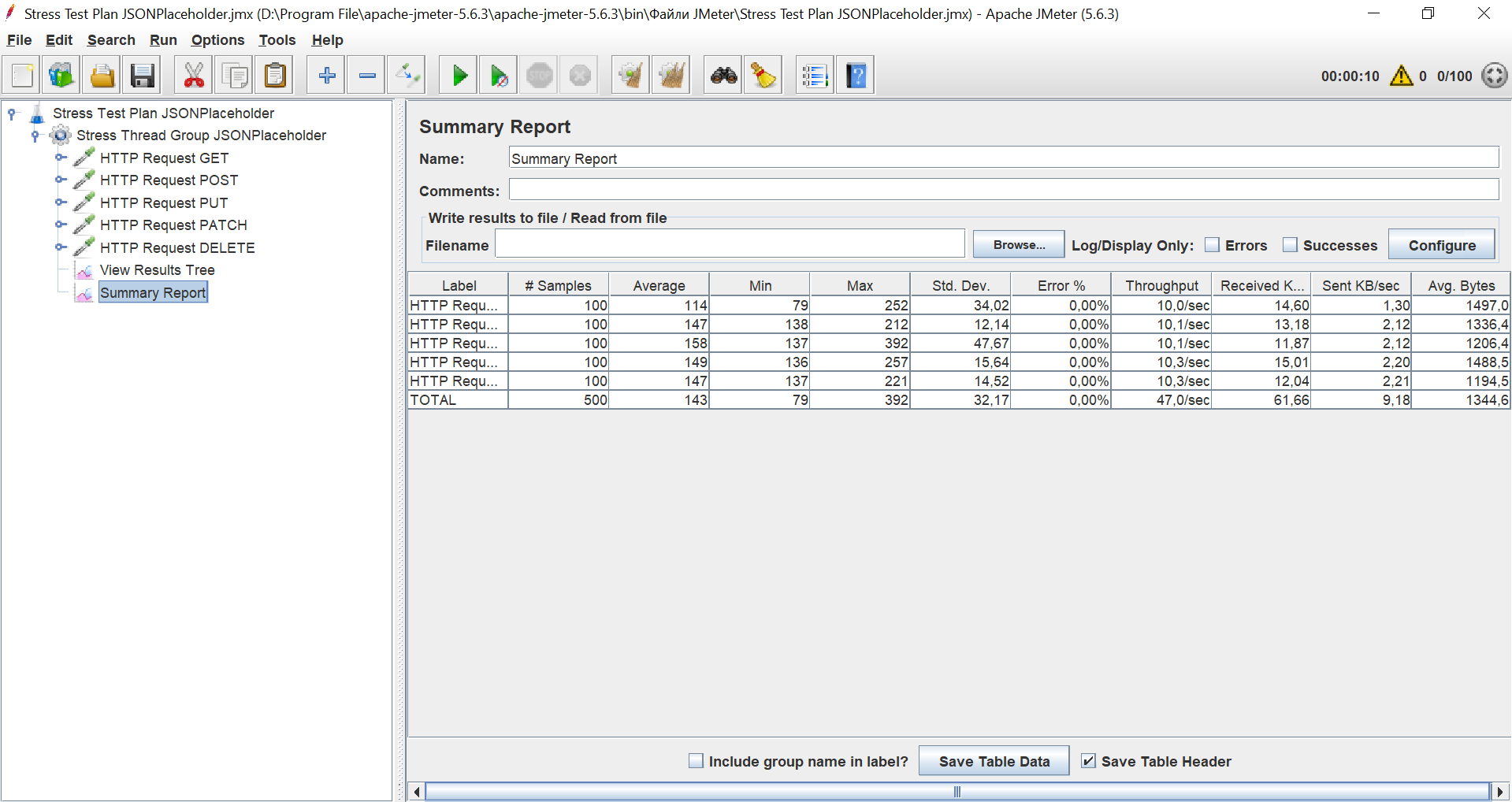


**Кількість користувачів = 100**

*View Results Tree*

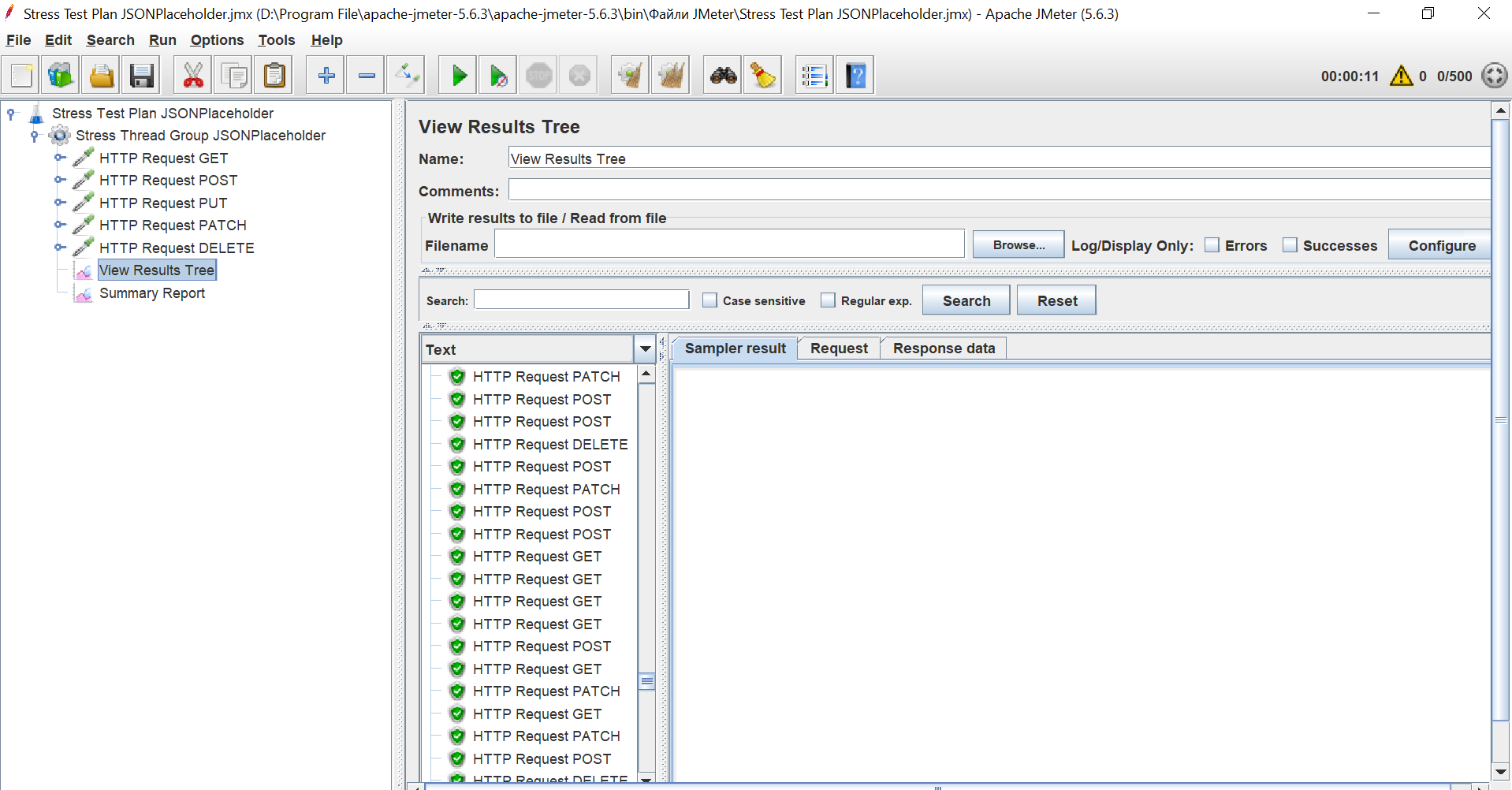


*Summary Report*

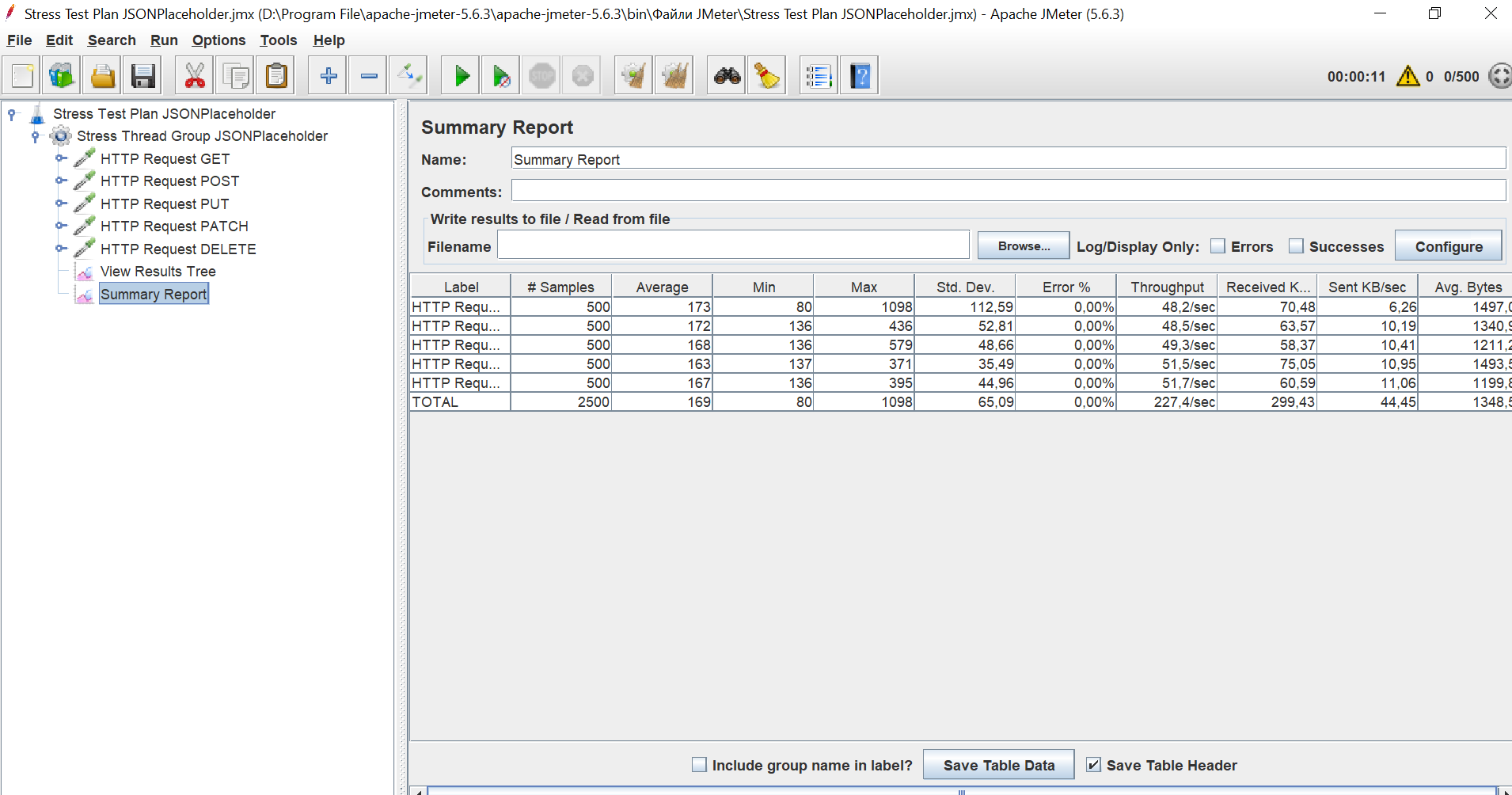


**Кількість користувачів = 500**

*View Results Tree*

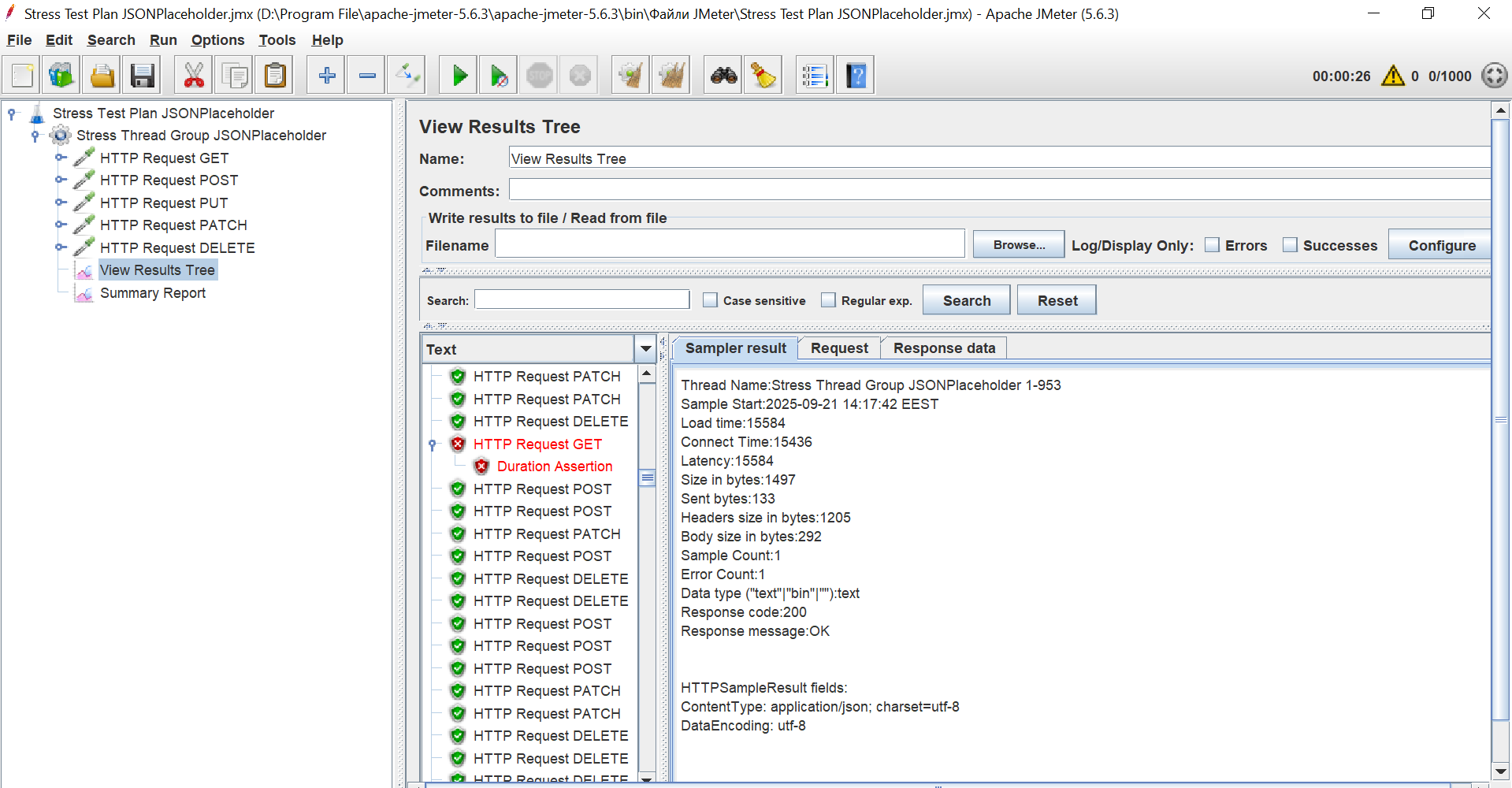


*Summary Report*

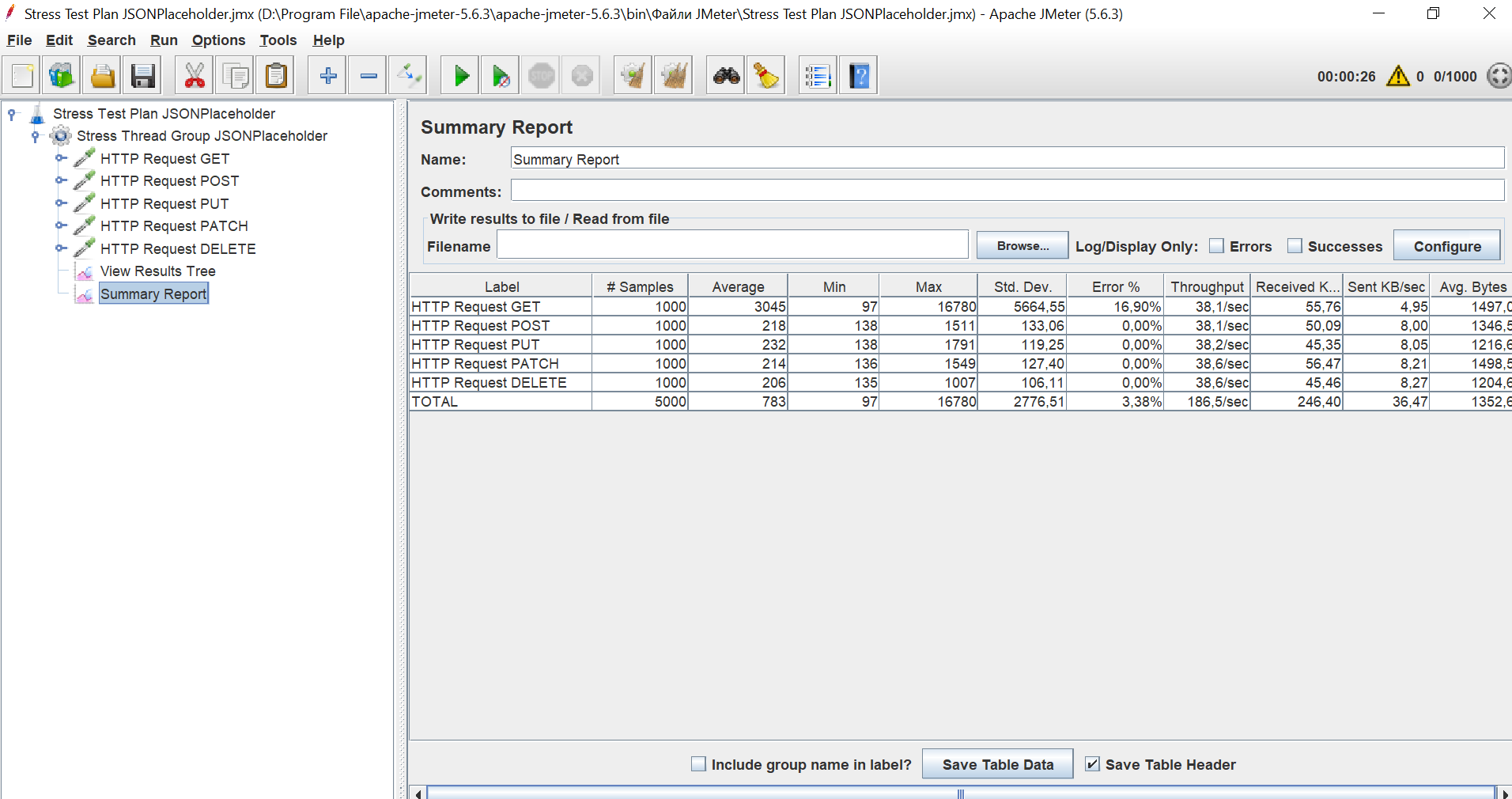


**Кількість користувачів = 1000**

*View Results Tree*



*Summary Report*



**Налаштування тесту:**

*Кількість користувачів: 1000*

*Ramp-up: 10 секунд*

*Тривалість: 60 секунд*

*Використані методи: GET, POST, PUT, PATCH, DELETE*

*Для кожного запиту були застосовані асерти: Response Code, Response Headers, Duration (<5000 мс), URL Assertion.*

*Лістенери: View Results Tree та Summary Report.*

**Результати виконання:**

*Запити POST, PUT, PATCH, DELETE залишились у зеленій зоні (усі асерти виконані успішно).*

*Запит GET не витримав навантаження:*

*Response Code = 200 (OK)*

*Проте Duration Assertion спрацював як помилка, оскільки фактичний час відгуку = 16 сек (> 5 сек).*

*Це вказує на суттєве зростання затримки при високій кількості користувачів.*

**Інтерпретація:**

*Система здатна обробляти одночасні запити до рівня ~500 користувачів без значних втрат продуктивності.*

*Починаючи з ~1000 користувачів, метод GET стає вузьким місцем, демонструючи надмірний час відгуку.*

*Інші методи під час тесту витримали навантаження, але прогнозовано можуть також деградувати при подальшому збільшенні трафіку.*

**Висновок:**

*Критична межа для цього API знаходиться у діапазоні між 500–1000 користувачів.*

*Метод GET потребує оптимізації.*

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

<https://github.com/AndreyMaslo4/GitLesson/tree/main/%D0%A1%D1%82%D1%80%D0%B5%D1%81%20%D0%BD%D0%B0%D0%B2%D0%B0%D0%BD%D1%82%D0%B0%D0%B6%D0%B5%D0%BD%D0%BD%D1%8F>

**3. Створи команду для запуску вказаного тест-плану через Bash CLI. Запусти її та завантаж результати в Summary Report.**

**Створи документ зі скриншотом виконання команди в командному рядку та скриншотом Summary Report.**

**Кількість користувачів = 500**

Admin@DESKTOP-2F0J49P MINGW64 /d/apache-jmeter-5.6.3/apache-jmeter-5.6.3/bin

$ #!/bin/bash

cd /d/apache-jmeter-5.6.3/apache-jmeter-5.6.3/bin

mkdir -p "/c/Users/Admin/GitLesson/Стрес навантаження/report"

java -jar ApacheJMeter.jar -n \

-t "/c/Users/Admin/GitLesson/Стрес навантаження/Stress Test Plan JSONPlaceholder 500 users.jmx" \

-l "/c/Users/Admin/GitLesson/Стрес навантаження/results.jtl" \

-e -o "/c/Users/Admin/GitLesson/Стрес навантаження/report"

WARN StatusConsoleListener The use of package scanning to locate plugins is depr

ecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is depr

ecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is depr

ecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is depr

ecated and will be removed in a future release

WARNING: A terminally deprecated method in sun.misc.Unsafe has been called

WARNING: sun.misc.Unsafe::objectFieldOffset has been called by com.thoughtworks.

xstream.converters.reflection.SunUnsafeReflectionProvider (file:/D:/apache-jmete

r-5.6.3/apache-jmeter-5.6.3/lib/xstream-1.4.20.jar)

WARNING: Please consider reporting this to the maintainers of class com.thoughtw

orks.xstream.converters.reflection.SunUnsafeReflectionProvider

WARNING: sun.misc.Unsafe::objectFieldOffset will be removed in a future release

Creating summariser <summary>

Created the tree successfully using C:/Users/Admin/GitLesson/Стрес навантаження/

Stress Test Plan JSONPlaceholder 500 users.jmx

Starting standalone test @ 2025 Sep 22 14:50:55 GMT+03:00 (1758541855597)

Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 44

45

summary + 656 in 00:00:04 = 155,5/s Avg: 475 Min: 141 Max: 2578 Err:

0 (0,00%) Active: 96 Started: 194 Finished: 98

summary + 1844 in 00:00:07 = 272,1/s Avg: 193 Min: 95 Max: 1663 Err:

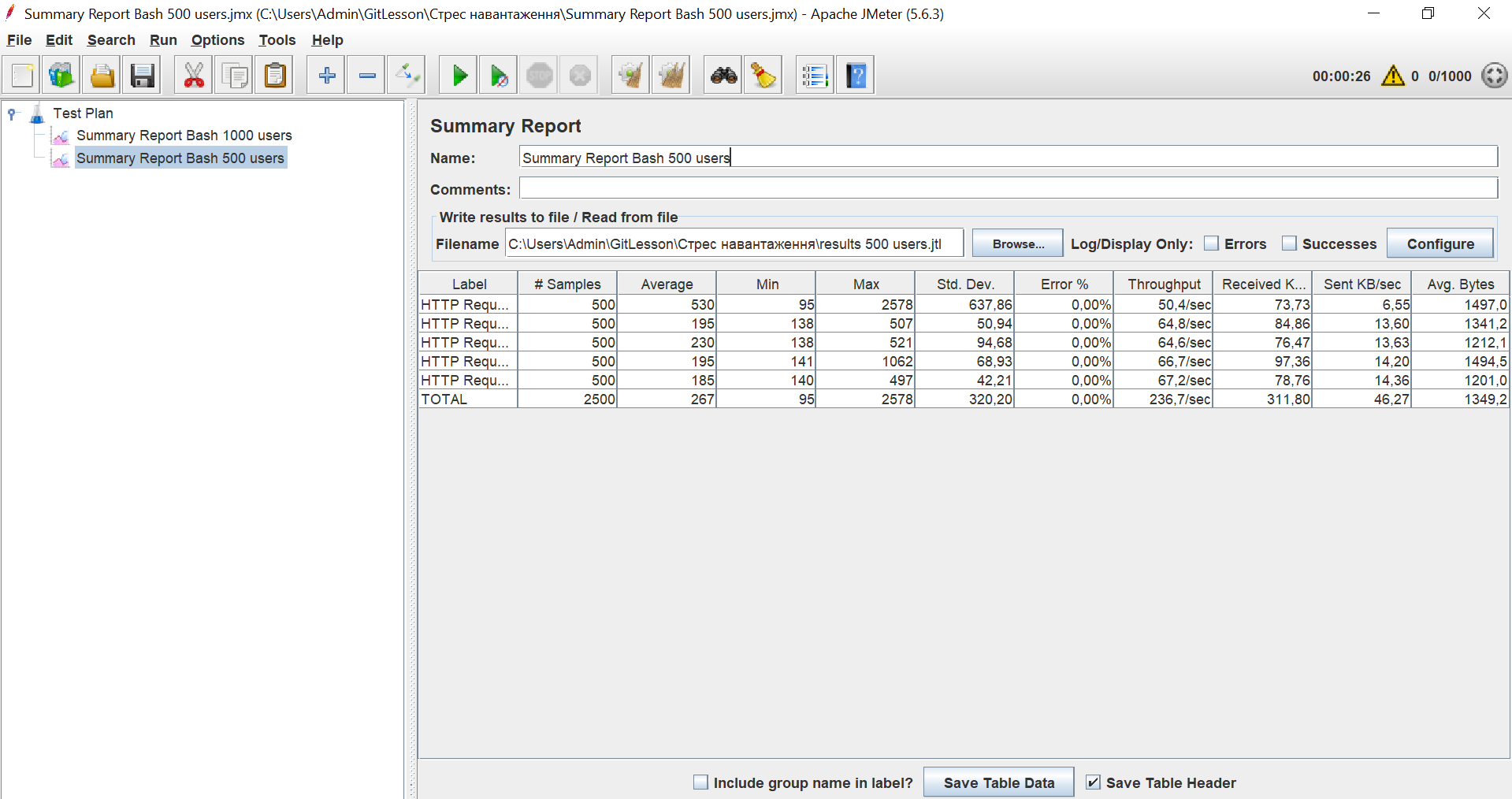
0 (0,00%) Active: 0 Started: 500 Finished: 500

summary = 2500 in 00:00:11 = 227,4/s Avg: 267 Min: 95 Max: 2578 Err:

0 (0,00%)

Tidying up ... @ 2025 Sep 22 14:51:06 GMT+03:00 (1758541866785)

... end of run



**Кількість користувачів = 1000**

Admin@DESKTOP-2F0J49P MINGW64 /d/apache-jmeter-5.6.3/apache-jmeter-5.6.3/bin

$ #!/bin/bash

cd /d/apache-jmeter-5.6.3/apache-jmeter-5.6.3/bin

mkdir -p "/c/Users/Admin/GitLesson/Стрес навантаження/report\_1000\_users"

java -jar ApacheJMeter.jar -n \

-t "/c/Users/Admin/GitLesson/Стрес навантаження/Stress Test Plan JSONPlaceholder 1000 users.jmx" \

-l "/c/Users/Admin/GitLesson/Стрес навантаження/results\_1000.jtl" \

-e -o "/c/Users/Admin/GitLesson/Стрес навантаження/report\_1000\_users"

WARN StatusConsoleListener The use of package scanning to locate plugins is depr

ecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is depr

ecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is depr

ecated and will be removed in a future release

WARN StatusConsoleListener The use of package scanning to locate plugins is depr

ecated and will be removed in a future release

WARNING: A terminally deprecated method in sun.misc.Unsafe has been called

WARNING: sun.misc.Unsafe::objectFieldOffset has been called by com.thoughtworks.

xstream.converters.reflection.SunUnsafeReflectionProvider (file:/D:/apache-jmete

r-5.6.3/apache-jmeter-5.6.3/lib/xstream-1.4.20.jar)

WARNING: Please consider reporting this to the maintainers of class com.thoughtw

orks.xstream.converters.reflection.SunUnsafeReflectionProvider

WARNING: sun.misc.Unsafe::objectFieldOffset will be removed in a future release

Creating summariser <summary>

Created the tree successfully using C:/Users/Admin/GitLesson/Стрес навантаження/

Stress Test Plan JSONPlaceholder 1000 users.jmx

Starting standalone test @ 2025 Sep 22 15:52:56 GMT+03:00 (1758545576109)

Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 44

45

summary + 436 in 00:00:04 = 116,8/s Avg: 705 Min: 102 Max: 2973 Err:

0 (0,00%) Active: 183 Started: 220 Finished: 37

summary + 4564 in 00:00:29 = 155,6/s Avg: 1329 Min: 117 Max: 23891 Err:

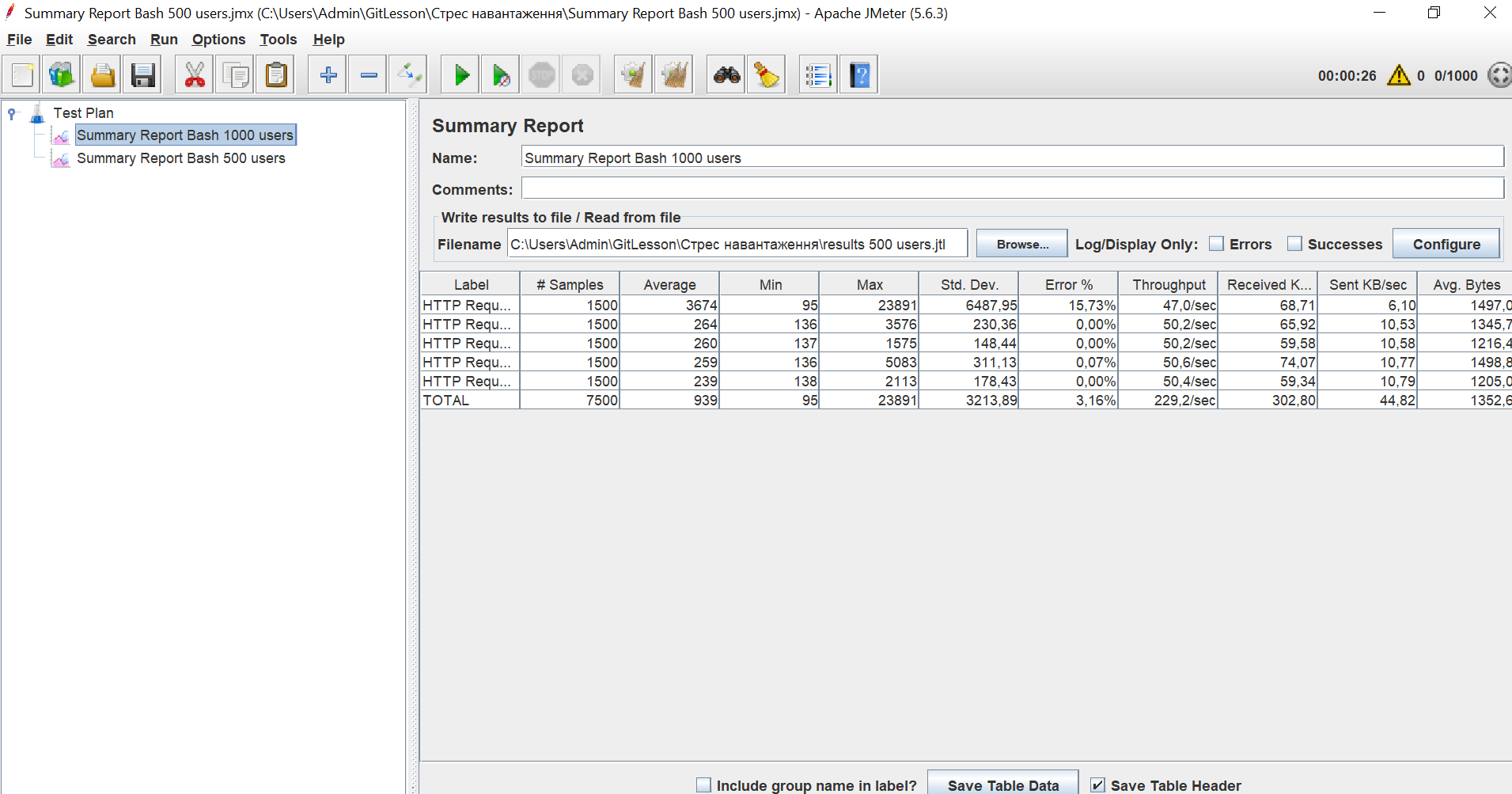
237 (5,19%) Active: 0 Started: 1000 Finished: 1000

summary = 5000 in 00:00:33 = 151,2/s Avg: 1275 Min: 102 Max: 23891 Err:

237 (4,74%)

Tidying up ... @ 2025 Sep 22 15:53:29 GMT+03:00 (1758545609343)

... end of run



**Документ додай у відповідний репозиторій в GitHub.**

[GitLesson/Стрес навантаження at main · AndreyMaslo4/GitLesson](https://github.com/AndreyMaslo4/GitLesson/tree/main/%D0%A1%D1%82%D1%80%D0%B5%D1%81%20%D0%BD%D0%B0%D0%B2%D0%B0%D0%BD%D1%82%D0%B0%D0%B6%D0%B5%D0%BD%D0%BD%D1%8F)