Lesson 24

**Load Testing Basics with Apache JMeter**

Level 1

**1. Create a set of GET, POST, PUT, PATCH, and DELETE requests to the JSONPlaceholder that are sent within 10 seconds in 3 iterations**

**2. Apply three different assertions to each of the requests.**

**3. The results of the tests should be received via the following Listeners:**

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

**4. Describe the conclusions of the test results (based on the reporters) in a separate file.**

**5. Add the created test plan and a document describing the results of the reporters' execution to your GitHub repository.**

Steps:

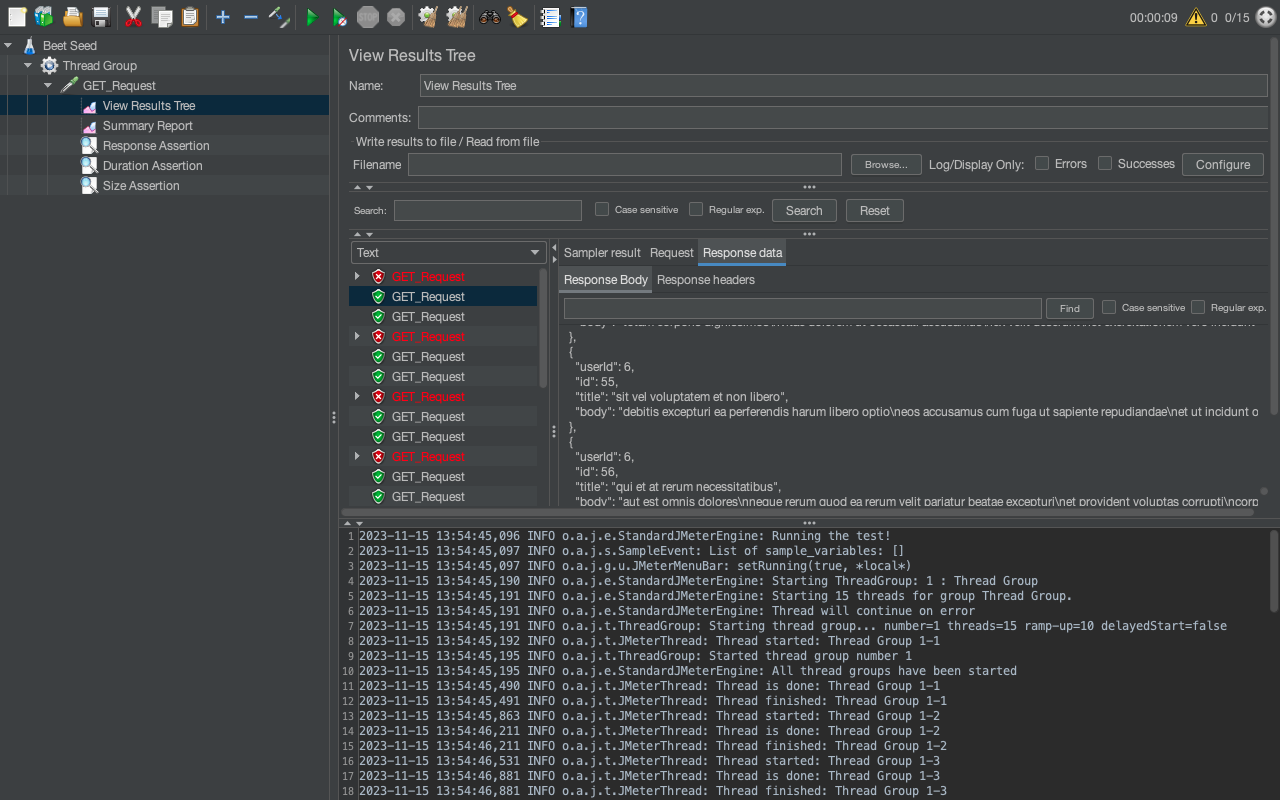
1. Create ‘Beet Seed’
2. Thread Group Setup
   1. Add a Thread Group - Practice
   2. Configuration:
      1. Number of Threads: 15 Users
      2. Ramp-up period: 10 sec
      3. Loop count: 3 iterations
3. Add Sampler
   1. HTTP Request
   2. Configure the Sampler with the appropriate method (GET, POST, PUT, PATCH, DELETE)
4. Add Listeners:
   1. View Results in Tree
   2. Summary Reports
5. Add Assertions
   1. Response Assertion
   2. Duration Assertion
   3. Size Assertion
6. Run Beet Seed

**GET Response**

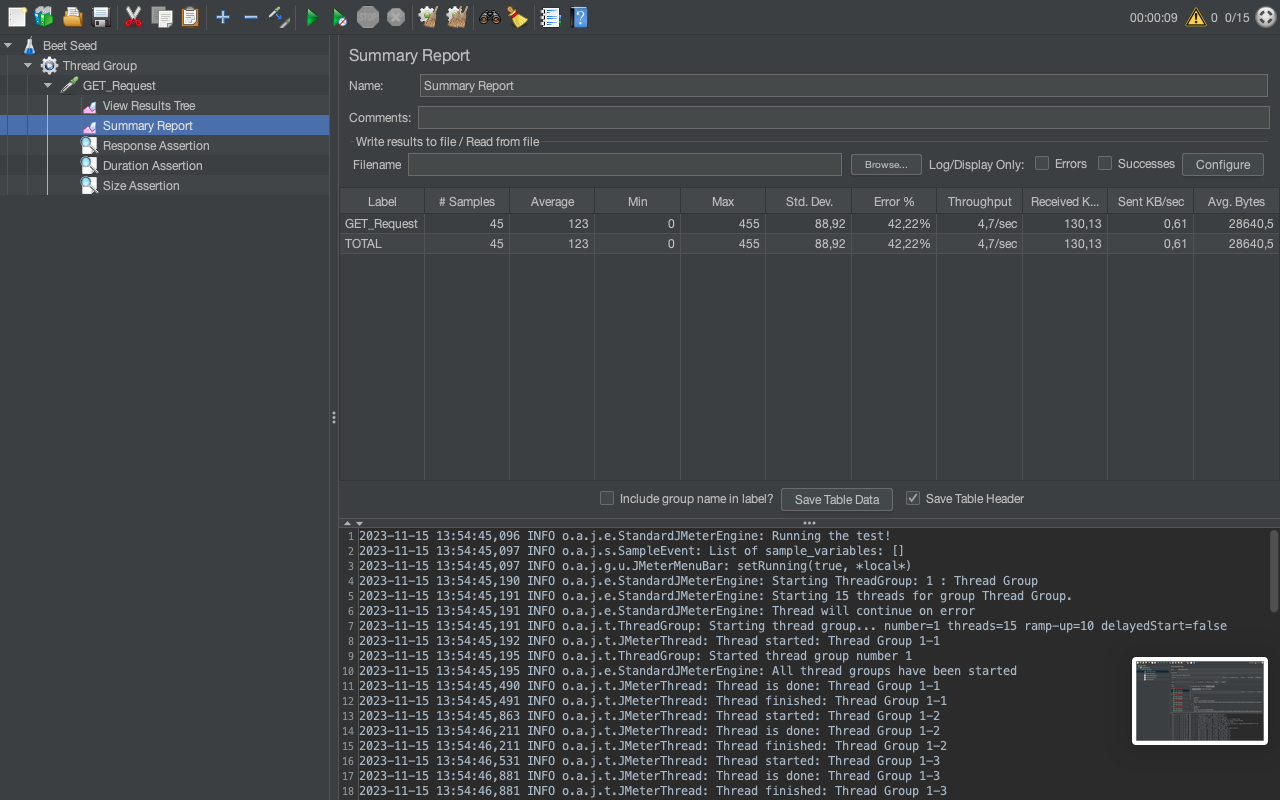
Assertion configuration:

* Response Assertion: - Response Code - Patterns to Test: 200
* Duration Assertion: - Duration in milliseconds: 150
* Size Assertion: - Size in Bytes >=28600

View Results Tree



Summary Report

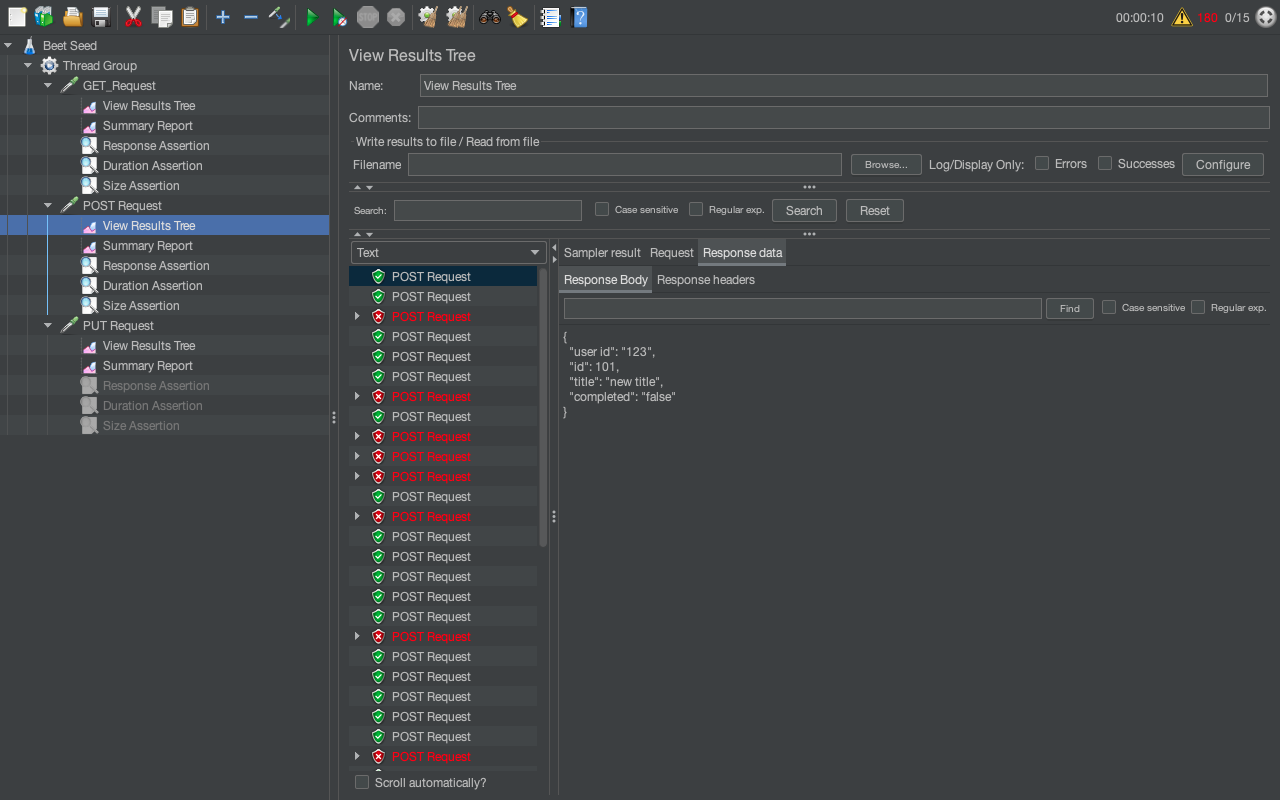


**POST Request**

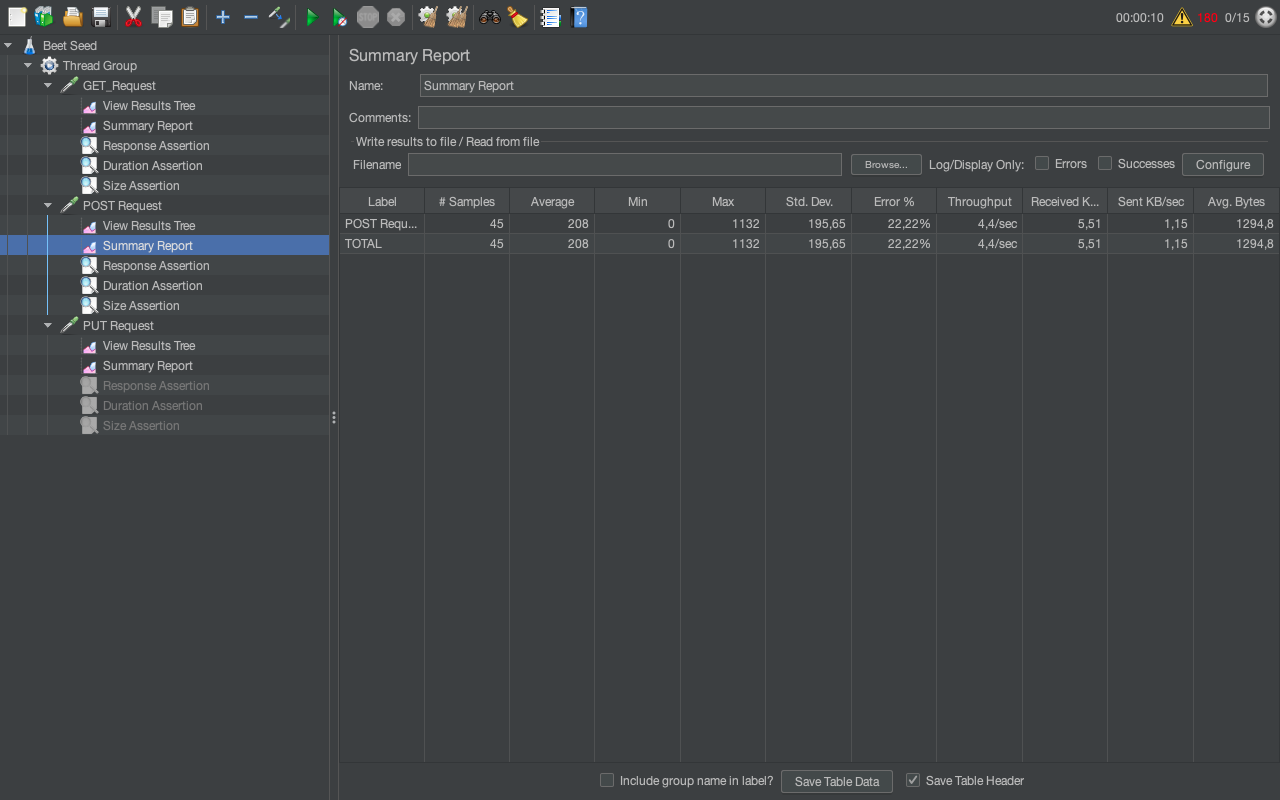
Assertion Configuration:

* Response Assertion: - Response Code - Patterns to Test: 201
* Duration Assertion: - Duration in milliseconds: 150
* Size Assertion: - Size in Bytes >=1200

View Results Tree



Summary Report

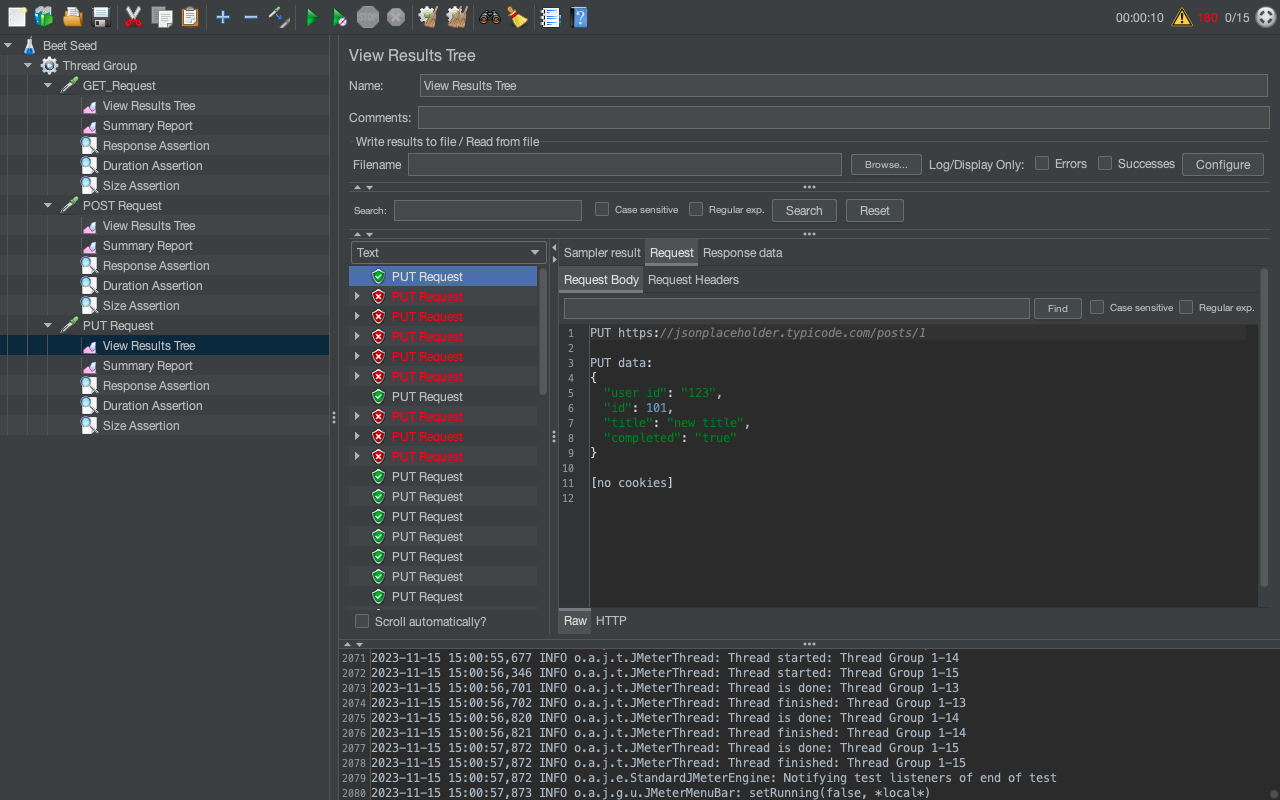


**PUT Request**

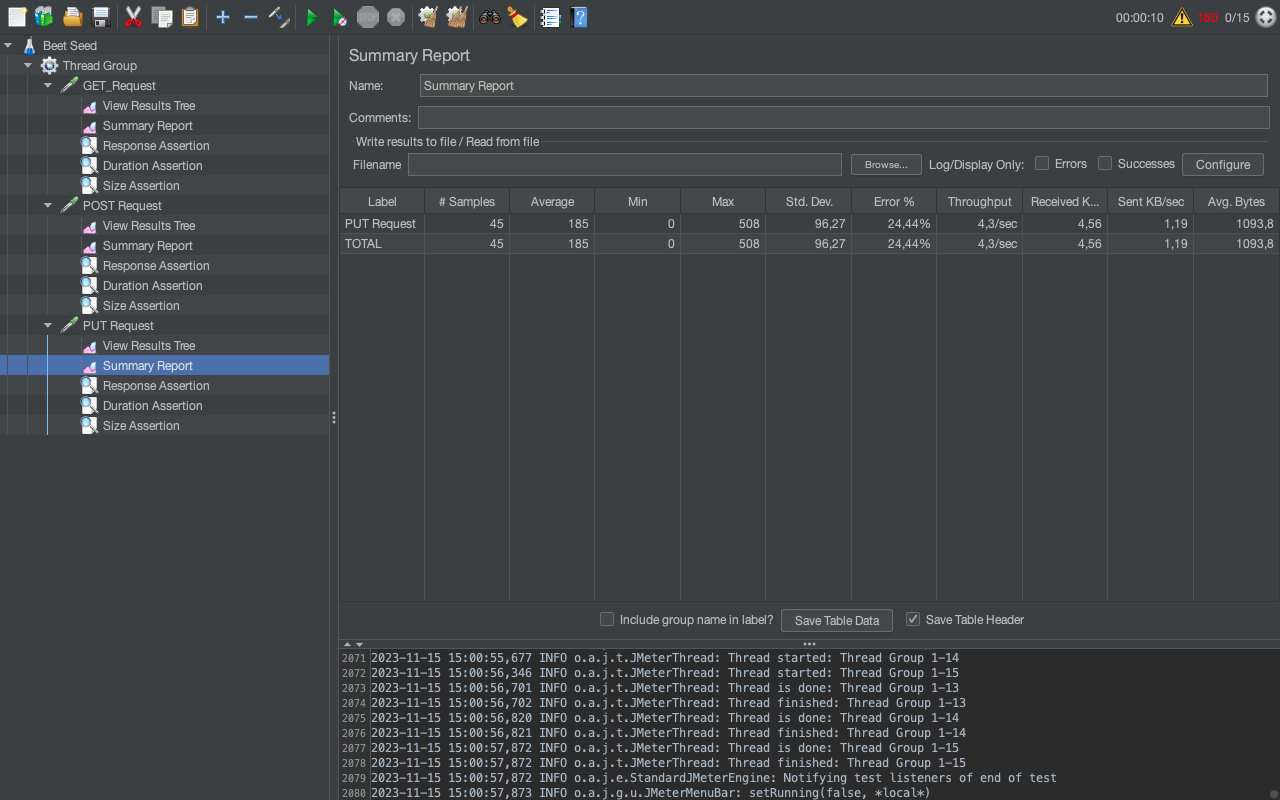
Assertion Configuration:

* Response Assertion: - Response Message - OK
* Duration Assertion: - Duration in milliseconds: 160
* Size Assertion: - Size in Bytes >=1000

View Results Tree



Summary Report

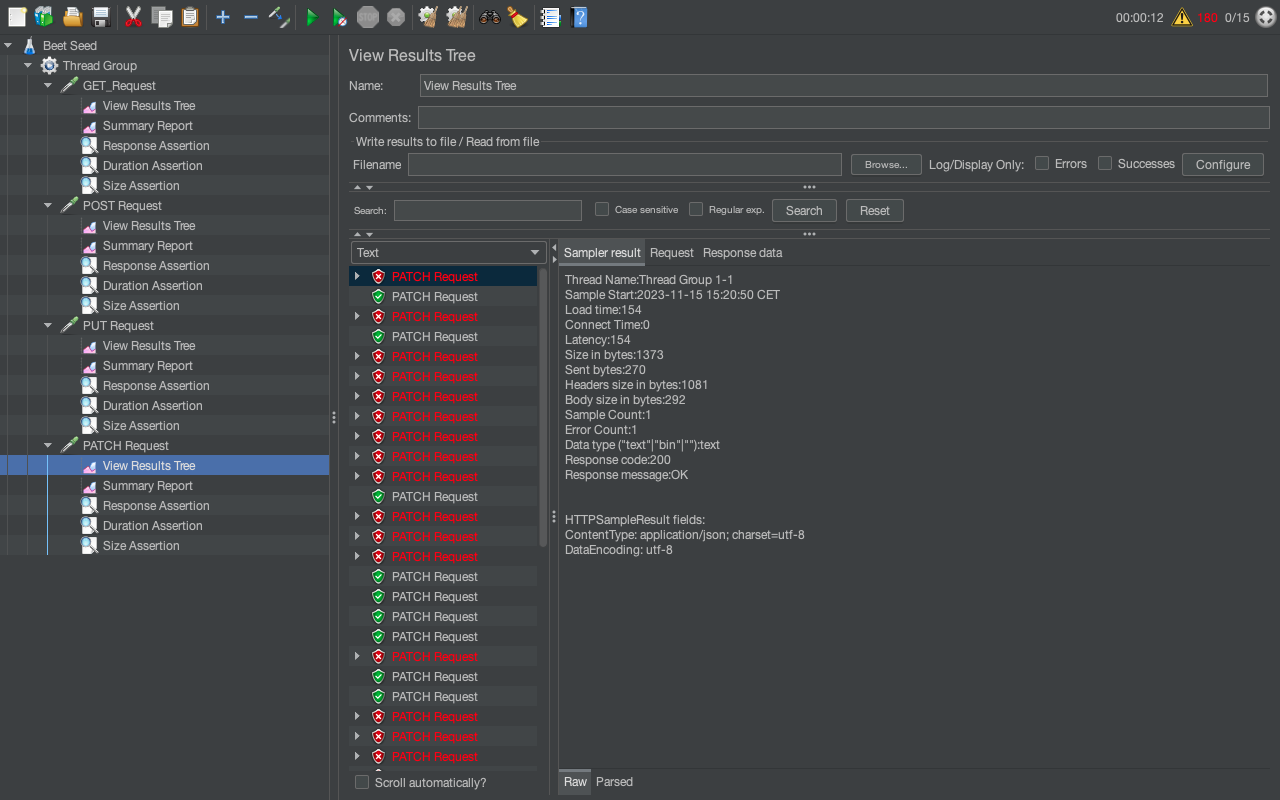


**PATCH Request**

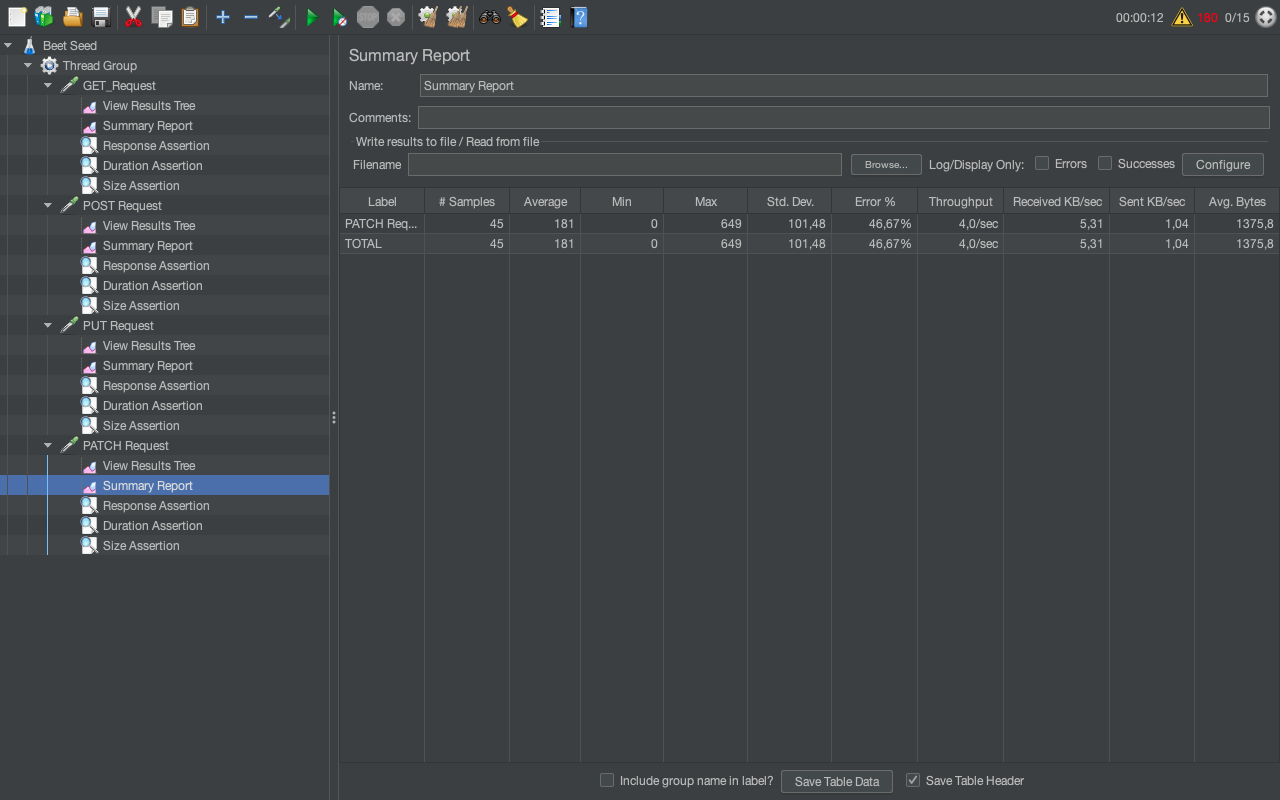
Assertion Configuration:

* Response Assertion: - Response Message - OK
* Duration Assertion: - Duration in milliseconds: 150
* Size Assertion: - Size in Bytes >=1300

View Results Tree



Summary Report

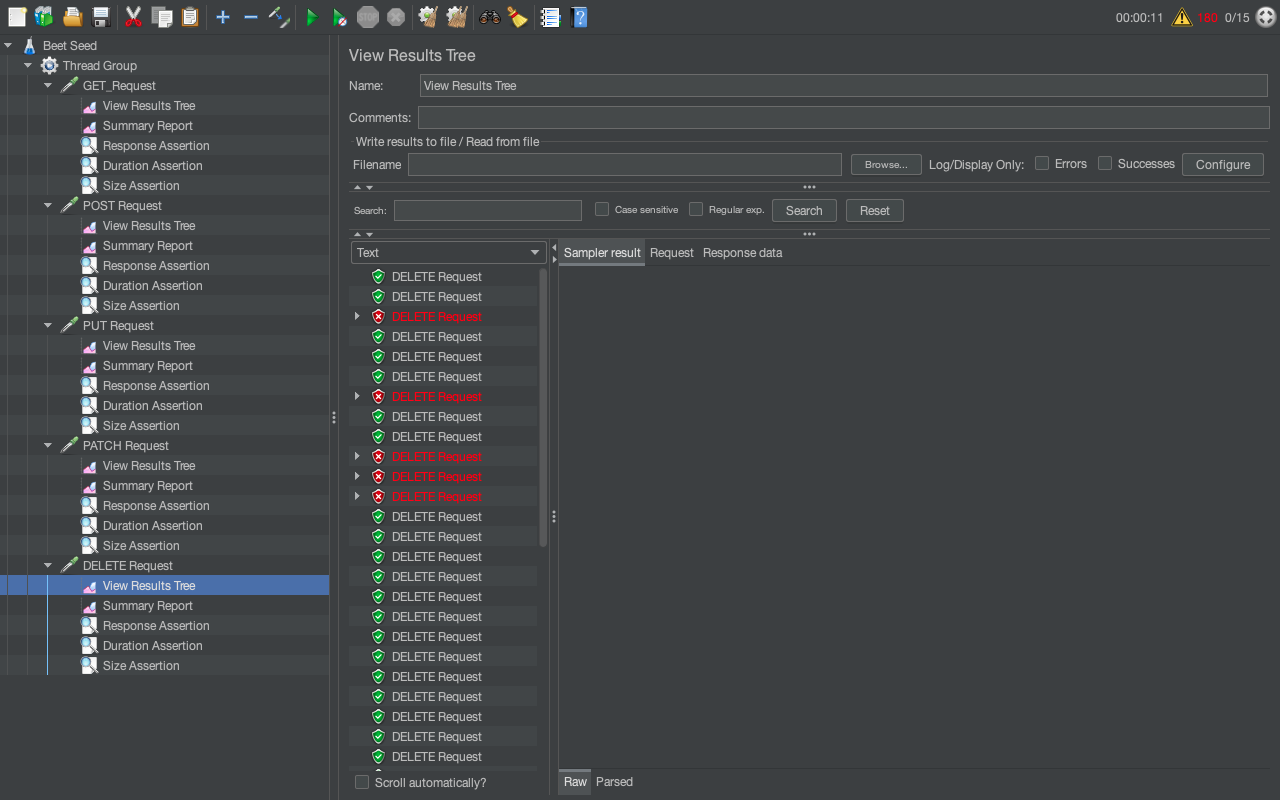


DELETE Request

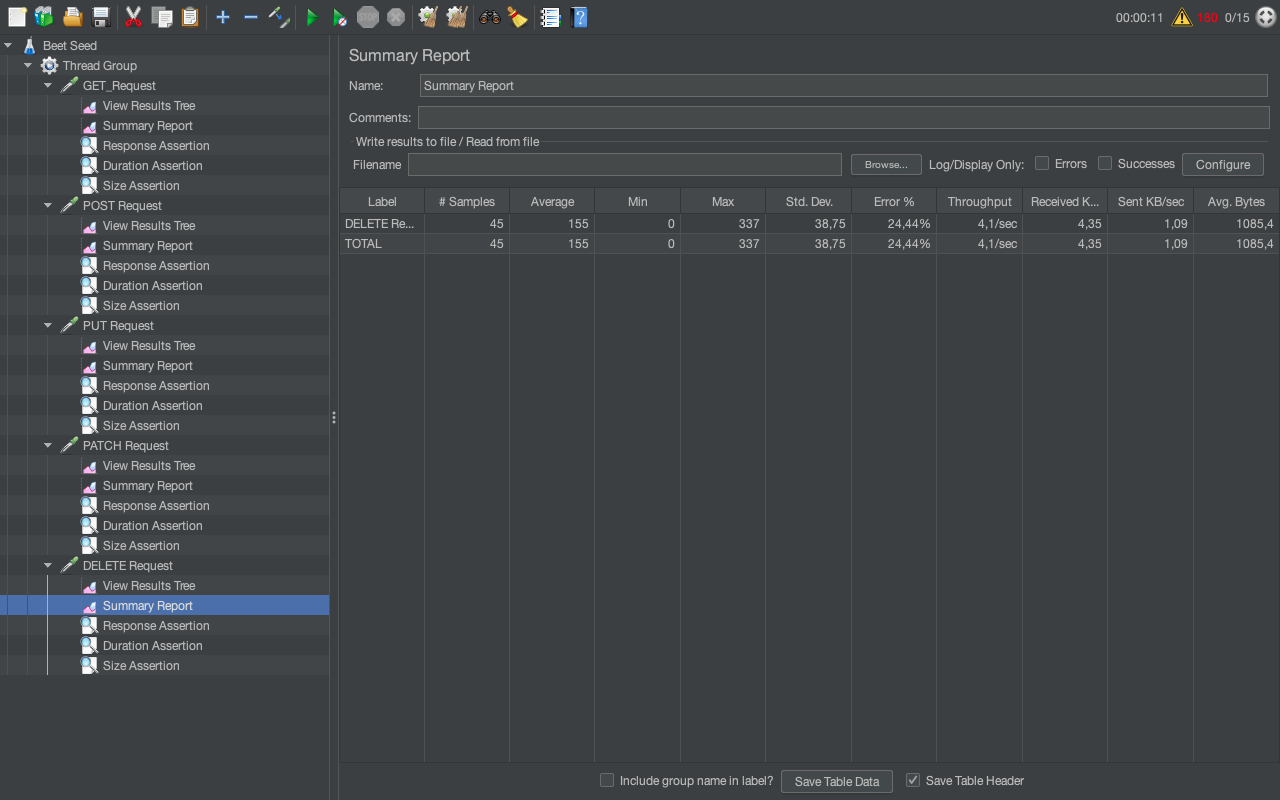
Assertion Configuration:

* Response Assertion: - Response Message - OK
* Duration Assertion: - Duration in milliseconds: 160
* Size Assertion: - Size in Bytes >=1081

View Results Tree



Summary Results



**Conclusions**

Test Results Analysis - JSONPlaceholder API

Summary of Findings:

- Overall, the API performed well with low average response times for GET requests.

- However, POST requests showed higher response time.

Performance Metrics:

GET: Throughput - 14,6 min; Error Rate: 39,93%

POST: Throughput - 15,5 min; Error Rate: 33,99%

PUT: Throughput - 9,7 min; Error Rate: 15,56%

PATCH: Throughput - 11,0 min; Error Rate: 31,11%

DELETE: Throughput - 2,1 sec; Error Rate: 24,44%

Issues Identified:

- Assertion failures observed in PATCH requests due to mismatched response data.

- Some DELETE requests resulted in '404 Not Found' errors.

Recommendations:

- Investigate and rectify PATCH request assertion failures.

- Address '404 Not Found' errors in DELETE requests by verifying resource existence before deletion.

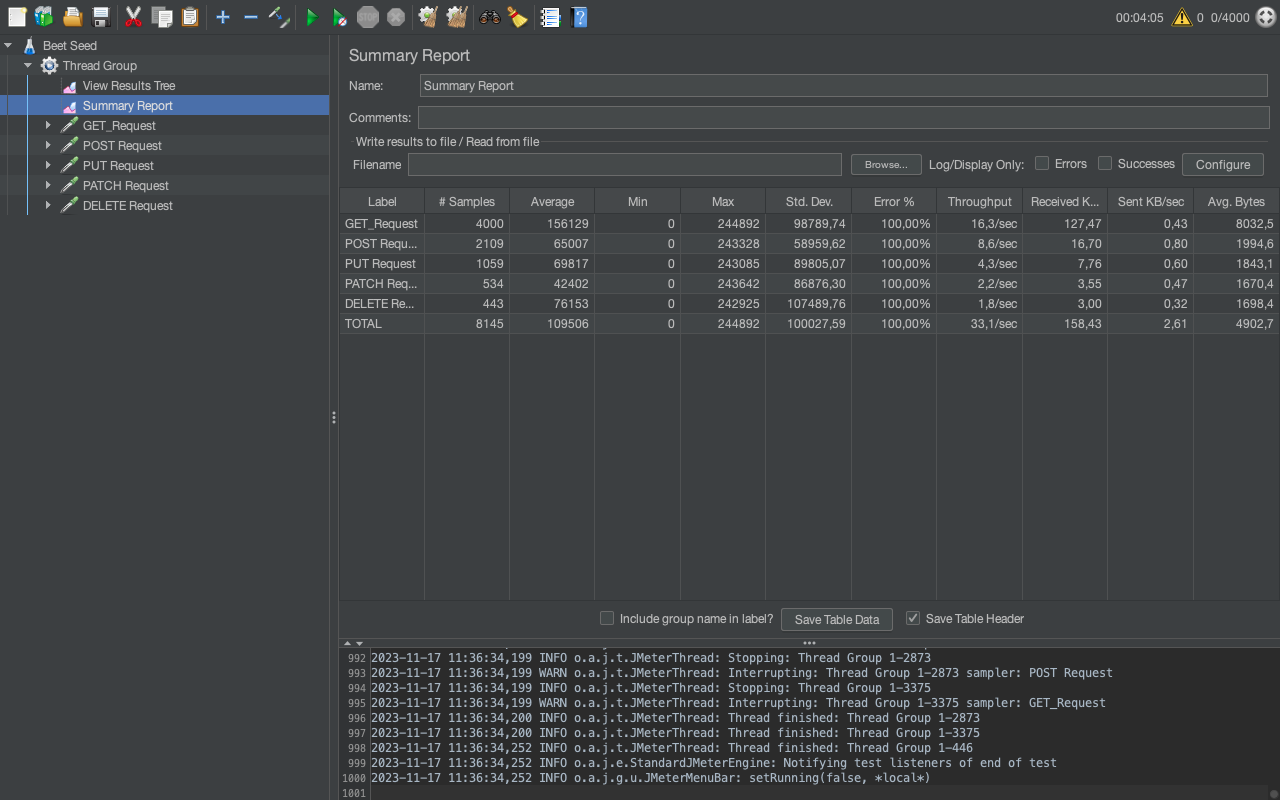
Level 2

**Create a new test plan based on one from the previous level. In it:**

**for each of the requests, perform a stress test of the used API;**

**identify the minimum combinations of parameters that will cause the web service to crash from the load.**

**Add the created test plan to your GitHub repository.**

****

**Conclusions**

This test aimed to assess the performance and identify the threshold at which the API experiences a crash or failure.

**Test Environment:**

Tool Used: Apache JMeter

Test Duration: [4 min 05 sec]

Tested API: [jsonplaceholder.typicode.com]

Concurrent Users: Increased gradually to [4000 users]

Parameters: Varied to stress-test the API with different combinations

**Test Methodology:**

The test plan involved simulating increased load by ramping up the number of concurrent users.

Parameters were systematically varied to assess combinations and their impact on the API.

JMeter's listeners (View Results Tree, Summary Report) were utilized to monitor and analyze the test results.

**Results Summary:**

Load:

Started with [10] users and increased gradually to [4000] users.

Response Times:

Average Response Time: [109506]

Maximum Response Time: [244892]