**JMeter Performance Testing Documentation**

**Project Name:** JSONPlaceholder API Performance Testing  
**Tool Used:** Apache JMeter  
**Assignment Task:** Performance Test Automation

**Table of Contents**

1. **Introduction**
2. **Testing Goals**
3. **JMeter Test Plan Setup**
   * **3.1 Test Scenario 1: Load Testing GET Endpoint**
   * **3.2 Test Scenario 2: Load Testing POST Endpoint**
4. **Execution and Results**
5. **Observations and Conclusion**

**1. Introduction**

This document provides a brief overview of the performance testing setup and results for the JSONPlaceholder API, focusing on two key scenarios: loading testing for a GET request and a POST request. The testing was conducted using Apache JMeter, a popular open-source tool for load testing and performance analysis.

**2. Testing Goals**

The main objectives of the performance tests were:

* **To evaluate the response times and throughput** of the JSONPlaceholder API under different load conditions.
* **To analyze the server behavior** when handling multiple concurrent requests.
* **To identify potential bottlenecks** that may affect performance.

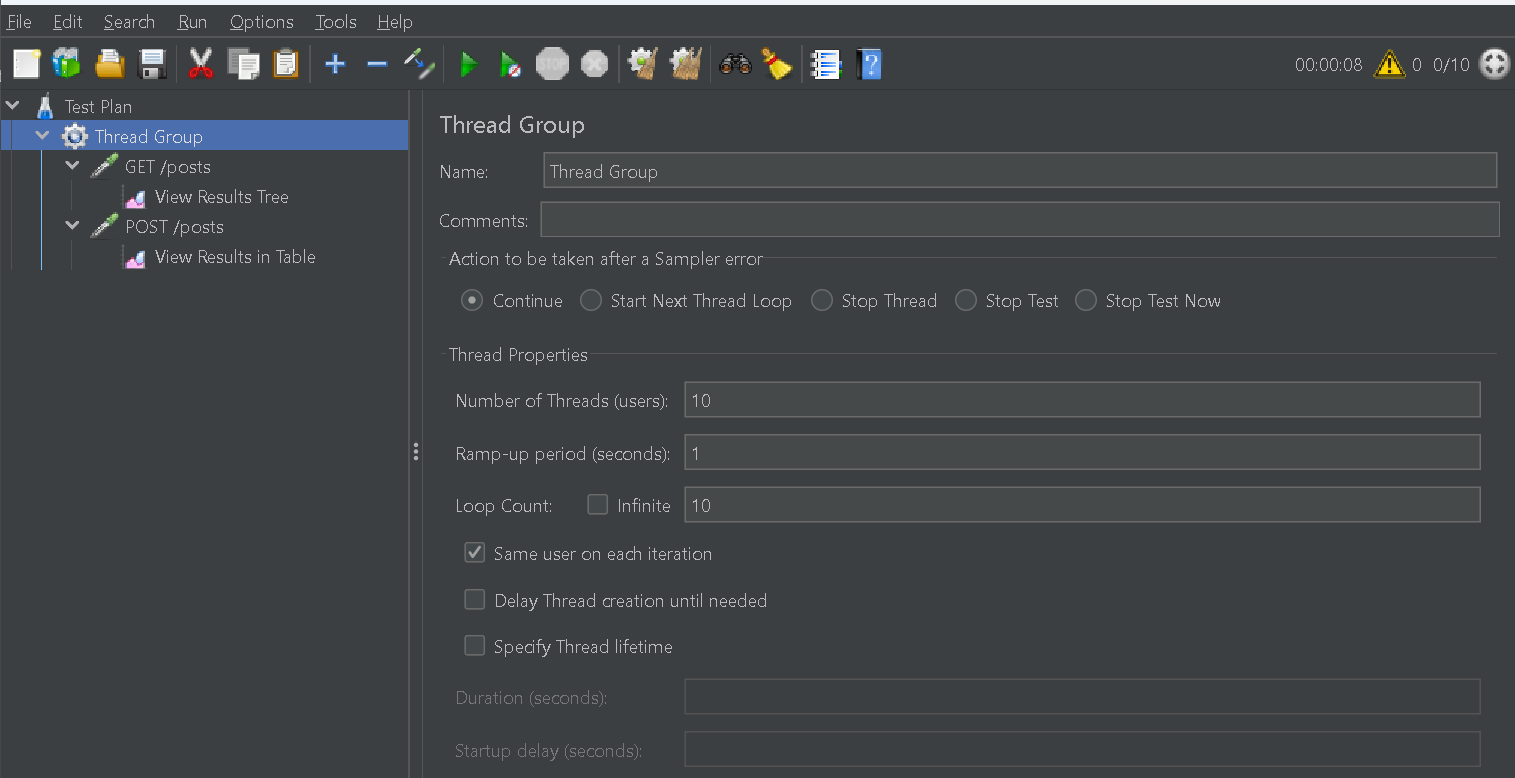
**3. JMeter Test Plan Setup**

**Tools and Configuration:**

* **Tool**: Apache JMeter
* **Thread Group Configuration**: Number of Threads (Users), Ramp-Up Period, and Loop Count for each test.

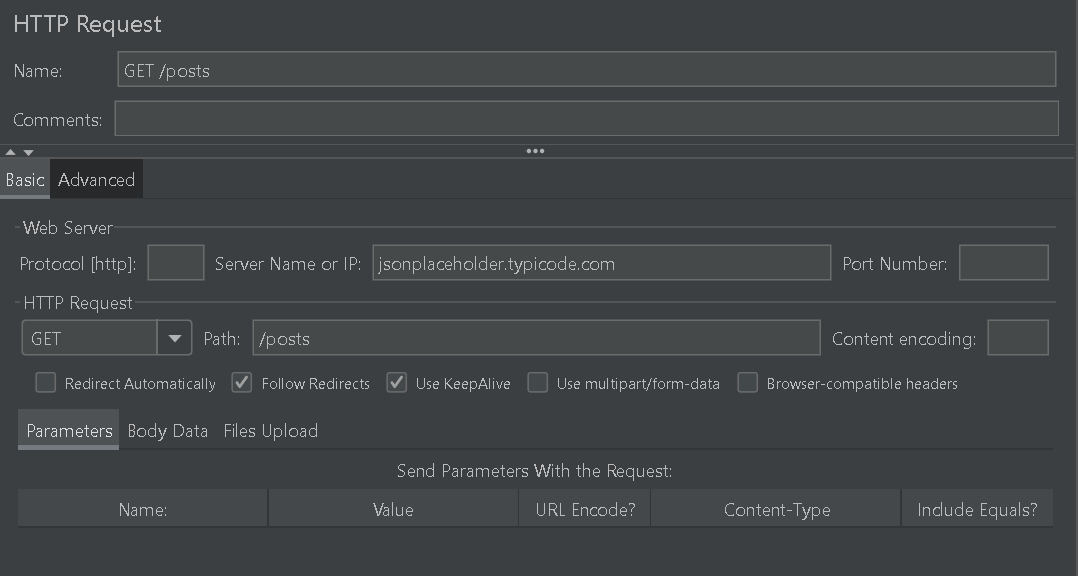
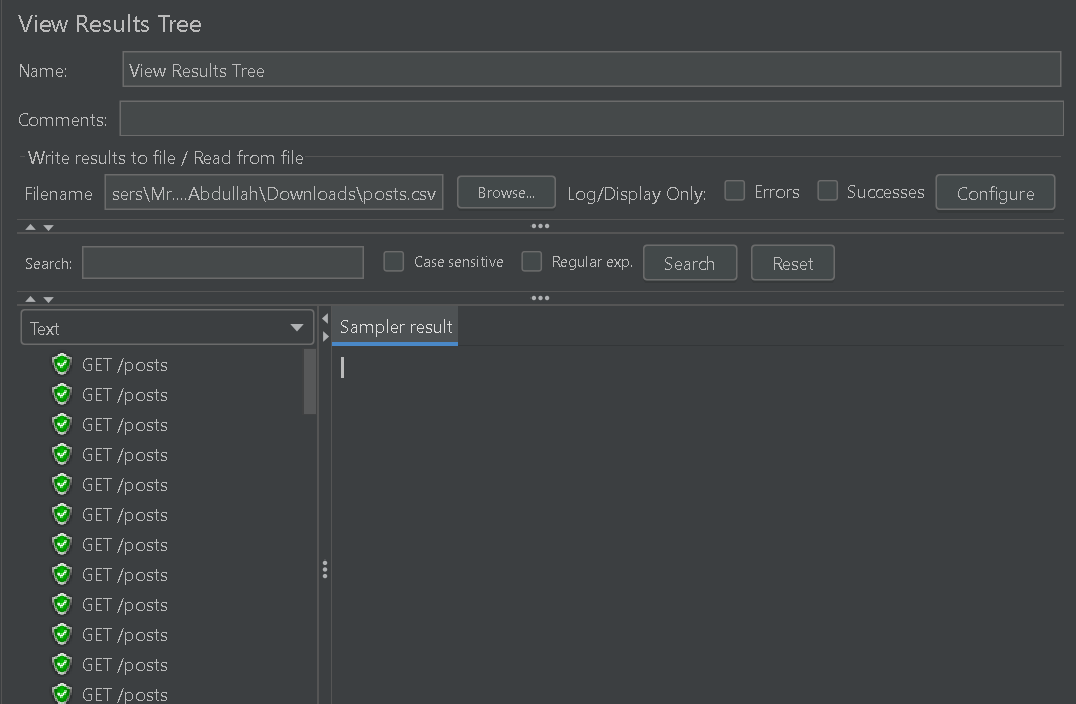
**Test Parameters:**

* **Target Server**: JSONPlaceholder API (https://jsonplaceholder.typicode.com)
* **Endpoints Tested**:
  + GET /posts
  + POST /posts



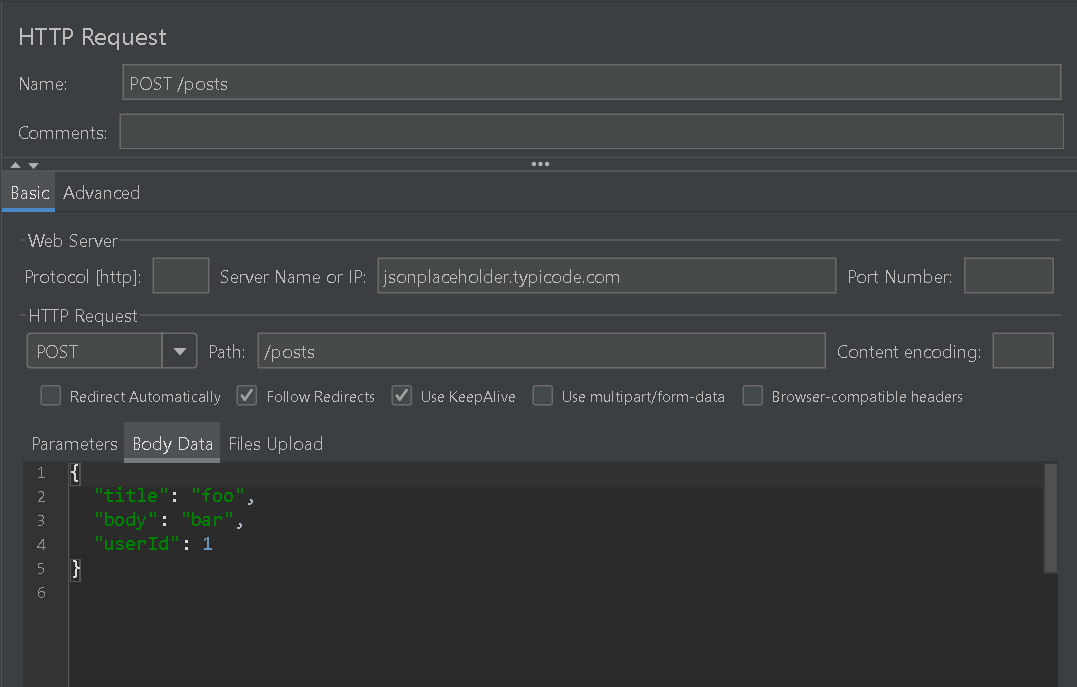
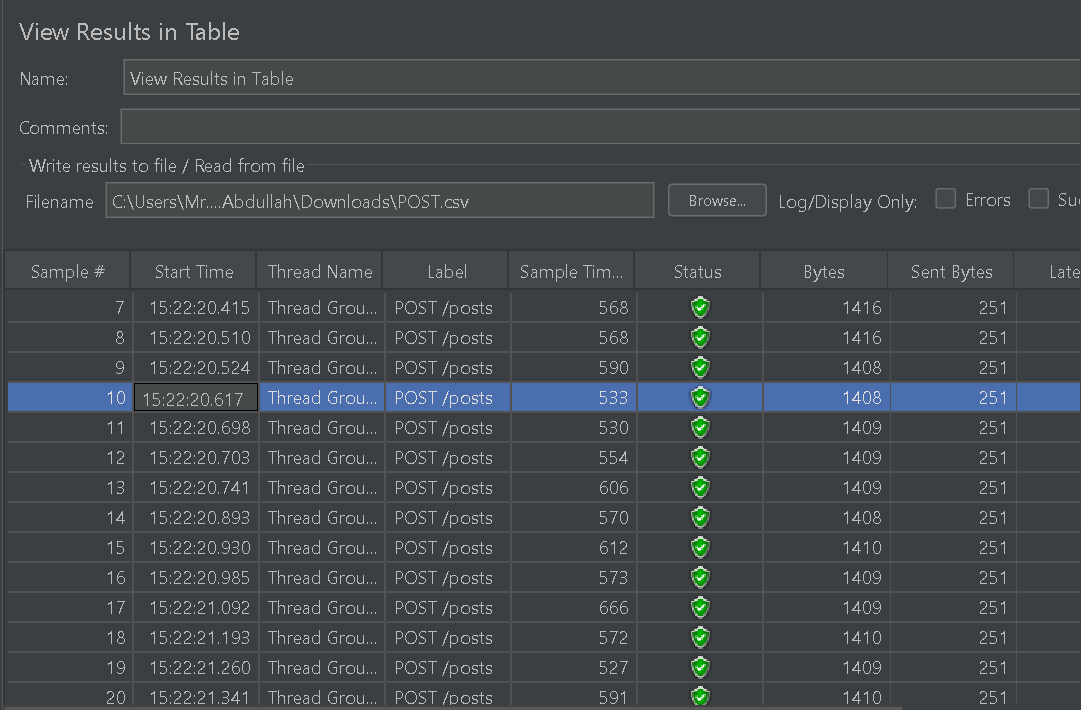
**3.1 Test Scenario 1: Load Testing GET Endpoint**

* **Description**: This test scenario was designed to evaluate the API's response time and throughput for the GET /posts endpoint.
* **Thread Group Settings**:
  + **Number of Threads**: 10
  + **Ramp-Up Period**: 10 seconds
  + **Loop Count**: 1 (single test per user)
* **Sampler**:
  + **HTTP Request**: Configured with the method GET and endpoint /posts.

**Screenshot 1**: Insert screenshot of your GET request setup in JMeter.  
  
  
  


**3.2 Test Scenario 2: Load Testing POST Endpoint**

* **Description**: This test scenario assesses the API's response under concurrent POST requests.
* **Thread Group Settings**:
  + **Number of Threads**: 10
  + **Ramp-Up Period**: 10 seconds
  + **Loop Count**: 1
* **Sampler**:
  + **HTTP Request**: Configured with the method POST and endpoint /posts, including JSON payload in the request body to simulate a new post creation.

**Screenshot 2**: Insert screenshot of your POST request setup in JMeter.  
  
  
  


**4. Execution and Results**

**4.1 Execution Process**

* **Step 1**: JMeter was launched, and each test scenario was configured in its own Thread Group.
* **Step 2**: Tests were executed by running the Thread Groups simultaneously, simulating multiple concurrent users accessing the endpoints.
* **Step 3**: Various listeners were added to gather performance metrics, including response time, throughput, and error rate.

**4.2 Results Summary**

**Screenshot 3**: Insert screenshots of response times and throughput graphs from JMeter for GET and POST requests.

**5. Observations and Conclusion**

Based on the results collected from JMeter:

* **Response Times**:
  + The GET endpoint maintained consistent response times even under high load.
  + The POST endpoint showed a slight increase in response time under load but remained within acceptable limits.
* **Throughput**:
  + Both endpoints handled the simulated load effectively, with minimal errors.
* **Conclusion**: JSONPlaceholder API performs efficiently for the tested GET and POST endpoints, handling the specified loads without performance degradation.