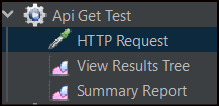
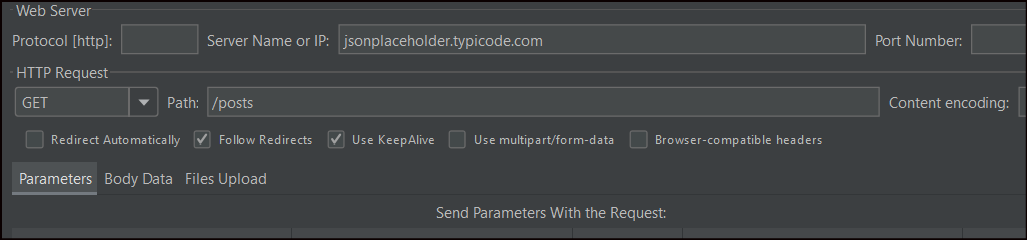
Assignment # 4

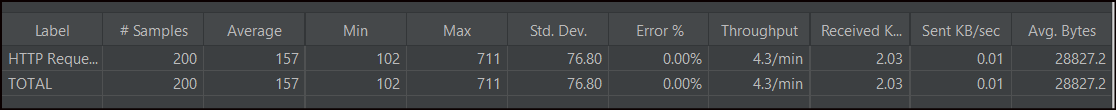
Test Plan using JMeter

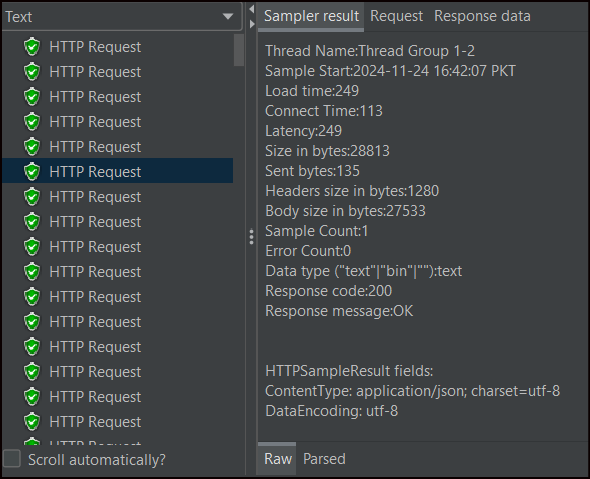
# Member1

# Api Get Test



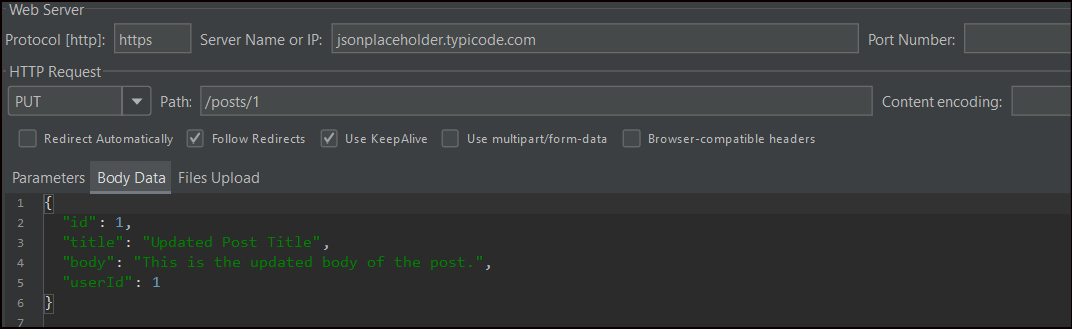


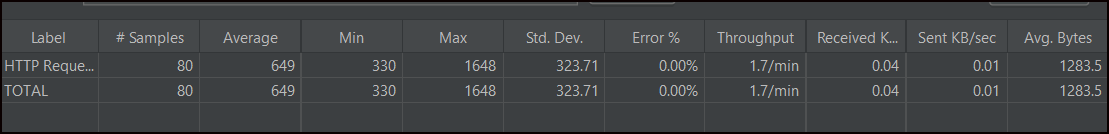


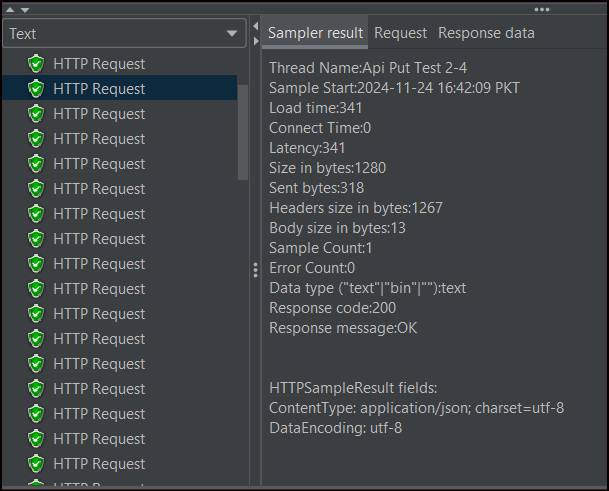


# Api Put Test



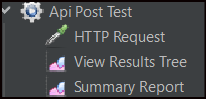


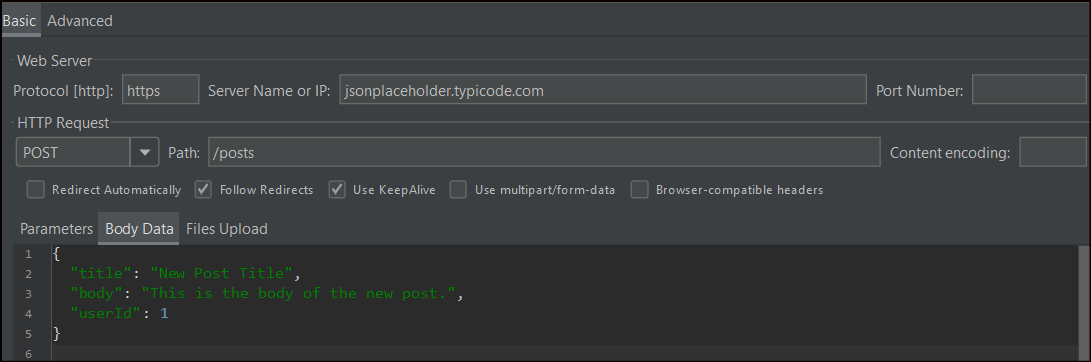


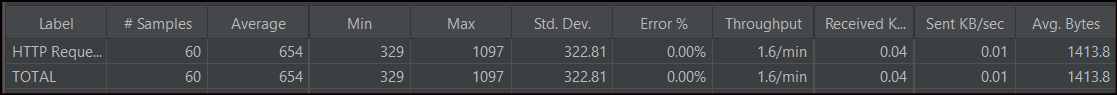


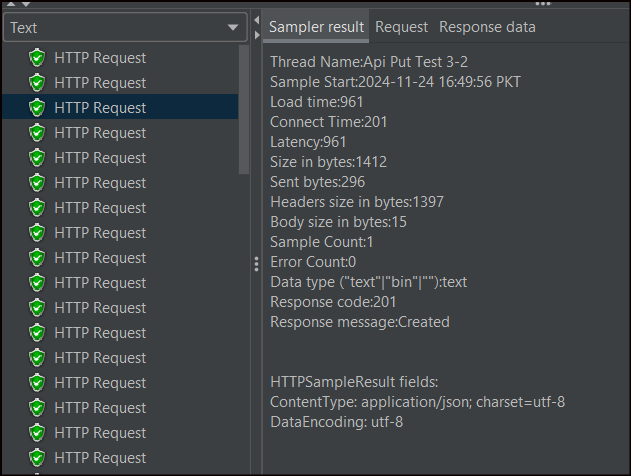
# Member2

# Api Post Test

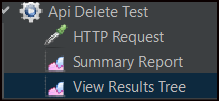


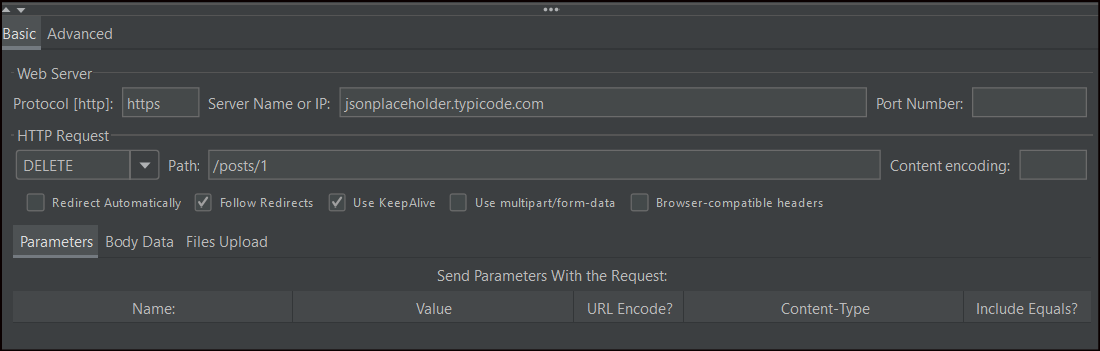


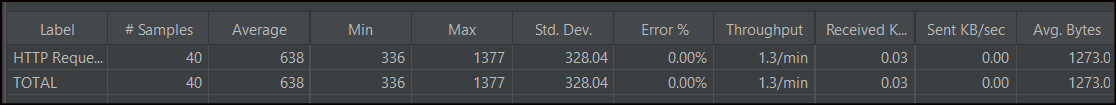


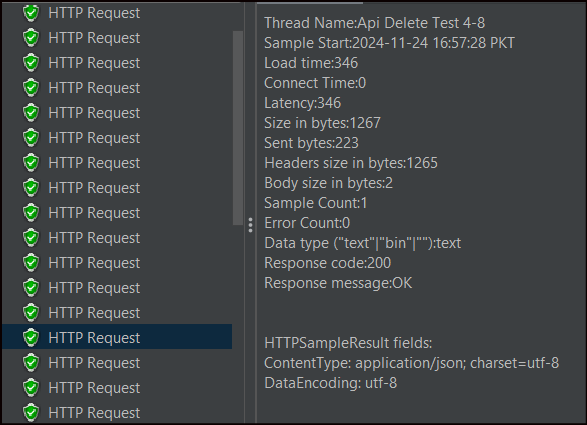
****

# Api Delete Test









# Task3

# Performance Requirements

For backend APIs, key performance benchmarks include:

* **Response Time**: The time taken to respond to a client request. Aim for under 200ms for typical APIs.
* **Throughput**: Number of requests handled per second. E.g., 1,000 requests/second for high-performance systems.
* **Scalability**: How the API performs as the load increases. Consider horizontal scaling (adding servers) and vertical scaling (increasing server resources).
* **Error Rate**: Percentage of failed requests. Aim for less than 0.1%.
* **Resource Utilization**: Memory, CPU, and bandwidth consumption under different loads.

**Tools to Explore:**

* JMeter: For load testing.
* Locust: For distributed load testing.
* Postman: For performance testing.

# Security Requirements for Restful APIs

Following the **OWASP REST API Security Checklist**, consider:

* **Authentication and Authorization**:
  + Use strong authentication mechanisms like OAuth 2.0.
  + Limit access with role-based permissions.
* **Data Validation**:
  + Validate inputs to prevent SQL injection or XSS attacks.
* **Secure Communication**:
  + Enforce HTTPS to encrypt data in transit.
* **Rate Limiting**:
  + Protect against DDoS attacks by limiting requests from a single IP.
* **Error Handling**:
  + Avoid exposing sensitive information in error messages.

**Security Tools:**

* OWASP ZAP (Zed Attack Proxy) for security scanning.
* Burp Suite for vulnerability assessments.

# Detailed Test Plan

**Key Components of a Test Plan:**

* **Objectives**:
  + Ensure APIs function as expected.
  + Verify performance benchmarks are met.
  + Identify and address vulnerabilities.
* **Testing Strategies**:
  + **Functional Testing**: Validate API endpoints against requirements.
  + **Performance Testing**: Simulate different loads and analyze performance.
  + **Security Testing**: Scan for vulnerabilities like SQL injection, CSRF, etc.
  + **Regression Testing**: Ensure new changes don’t break existing functionality.
* **Test Tools**:
  + **Postman**: For functional and regression testing.
  + **JMeter**: For load testing.
  + **New Relic**: For monitoring performance.
  + **OWASP ZAP**: For security testing.
* **Automation**:
  + Use tools like Rest Assured (Java-based library for API testing) or Postman Collections for automated testing.