

Automating Reqres APIs using Postman RestAssured JMeter

Table of Contents

<i>Introduction</i>	- 1 -
<i>Problem Statement</i>	- 2 -
<i>Requirements Detail</i>	- 2 -
<i>Solution Details</i>	- 2 -
<i>Project Structure and Results – Postman</i>	- 3 -
Test Run Collection	- 3 -
Test Run Result Summary	- 3 -
Test Run Details	- 3 -
Test Run Result on Console	- 4 -
<i>Project Structure and Results – JMeter</i>	- 4 -
CRUD Test JMX	- 4 -
Load Test Execution Report.....	- 5 -
Stress Test Execution Report.....	- 5 -
Spike Test Execution Report.....	- 5 -
<i>Project Structure and Results – Rest Assured</i>	- 6 -
Log4j Property File	- 6 -
Code Snippet	- 6 -
Log Output.....	- 7 -

Introduction

- This is the project report for the Project 1: **Testing the CRUD Operations and performance of List of Users** for “Test and monitor your applications through Non-functional and API Testing” course of Simplilearn.
- Project submitted by : Krishnaveni Rajan
- Email Id: krisnnaveni07.rajan@gmail.com
- Submission Date: 03-November-2021
- Language/Tools Used : Postman, JMeter, Rest Assured
- CI/CD : GitHub

Problem Statement

Given an API which has details about Users of an Organization. Test CRUD operations of this API using Postman and automate using Rest Assured and validate the response. Check the performance of an API using JMeter.

As a part of developing an API, Automate and Check about the Users Details.

Requirements Detail

1. Send a GET request using Postman to API and check list of users
2. Validate the Status codes generated for the above requests
3. Sending a POST request using Postman to API to add user and check the response.
4. Write automation script for GET Request using Rest Assured
5. Write automation script for POST Request using Rest Assured
6. Generate Log on Eclipse Console using Log4j
7. Validate the Response
8. Check the above web services performance using JMeter
9. Do Load Testing.
10. Do Stress Testing.
11. Do Spike Testing.

Solution Details

Solution for this project is provided in 3 different tools.

- Requirement #1 to #3 is addressed using Postman.
- Requirement #4 to #7 is addressed using Rest Assured.
- Requirement #8 to #11 is addressed using JMeter.

Project Structure and Results – Postman

5 different requests has been created for the CRUD operation using methods GET, POST, PUT and DELETE.

Test Run Collection

Test Run Result Summary

Test Run Details

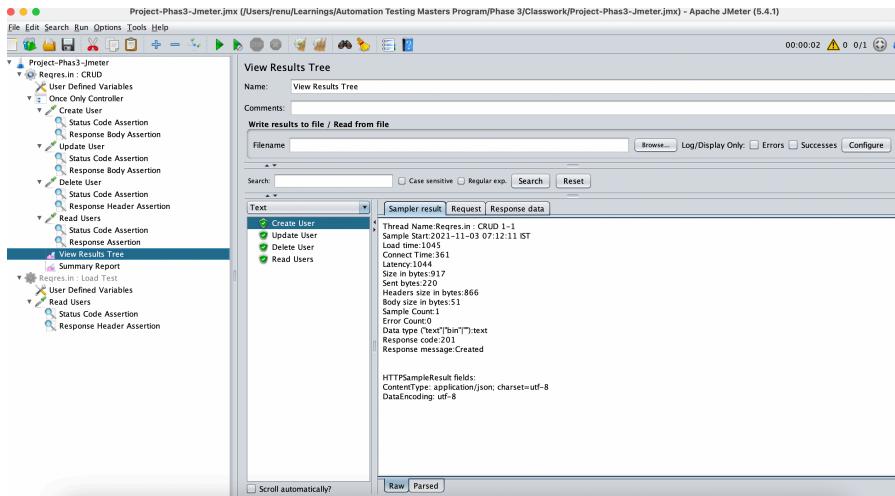
Test Run Result on Console

Project Structure and Results – JMeter

Multiple threads were created in the JMeter. 1st Thread was created for CRUD operation is built with ‘Once only controller’ where each request is run only once to meet Create, Read, Update and Delete operation. Assertions are used to verify the Status Code and content.

Rest of the threads are created for Load, Stress and Spike testing for the List users API which is of GET request.

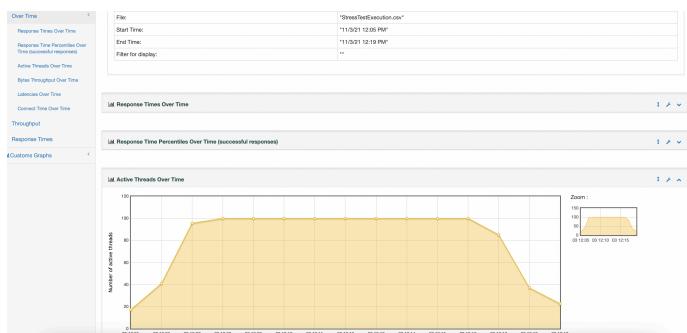
CRUD Test JMX



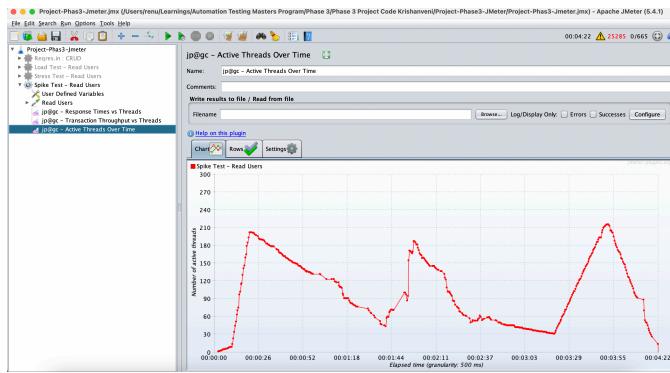
Load Test Execution Report



Stress Test Execution Report



Spike Test Execution Report



Project Structure and Results – Rest Assured Log4j Property File

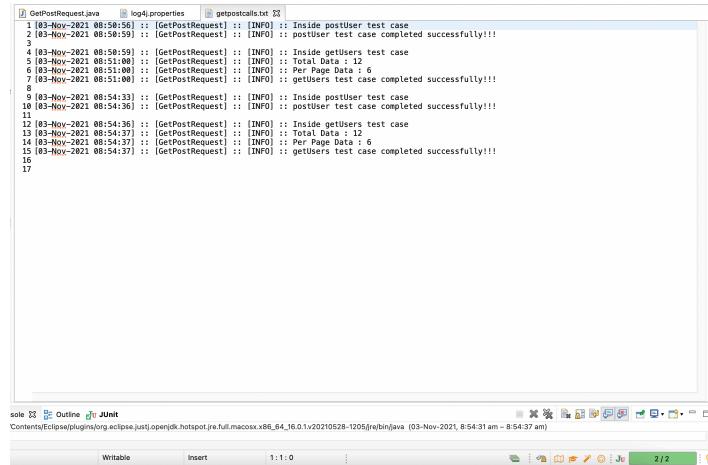
```
1 log4j.rootLogger=INFO,fa
2
3 #File Appender
4 log4j.appender.fa=org.apache.log4j.RollingFileAppender
5 log4j.appender.file.MaxFileSize=1MB
6 log4j.appender.file.MaxBackupIndex=1
7 log4j.appender.fa.File=logs/getpostcalls.txt
8 log4j.appender.fa.append=true
9 log4j.appender.fa.layout=org.apache.log4j.PatternLayout
10 log4j.appender.fa.layout.ConversionPattern=[%d{dd-MMM-YYYY hh:mm:ss}] :: [%c] :: [%p] :: %m %n
```

Code Snippet

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** myworkspace - Project-Phase3-RestAssured/src/test/java/Reqres/GetPostRequests.java - Eclipse IDE
- Code Editor:** The main window displays Java code for a REST API test. The code uses RestAssured to make requests to a base URL (`https://reqres.in/`). It includes methods for listing users, getting a user by ID, creating a user, and updating a user.
- Toolbars:** Standard Eclipse toolbars for file operations, search, and navigation.
- Bottom Status Bar:** Shows "Runs: 2/2", "Errors: 0", "Failures: 0", and a "Failure Trace" button.

Log Output



The screenshot shows the Eclipse IDE interface with the 'Log' view open. The log output is displayed in a text area, showing the execution of a JUnit test named 'GetPostRequest.java'. The log entries are as follows:

```
1 [03-Nov-2021 08:50:56] :: [GetPostRequest] :: [INFO] :: Inside postUser test case
2 [03-Nov-2021 08:50:56] :: [GetPostRequest] :: [INFO] :: postUser test case completed successfully!!!
3 [03-Nov-2021 08:50:59] :: [GetPostRequest] :: [INFO] :: Inside getUsers test case
4 [03-Nov-2021 08:51:00] :: [GetPostRequest] :: [INFO] :: Total Data : 12
5 [03-Nov-2021 08:51:00] :: [GetPostRequest] :: [INFO] :: Per Page Data : 6
6 [03-Nov-2021 08:51:00] :: [GetPostRequest] :: [INFO] :: getUsers test case completed successfully!!!
7 [03-Nov-2021 08:54:33] :: [GetPostRequest] :: [INFO] :: Inside postUser test case
8 [03-Nov-2021 08:54:33] :: [GetPostRequest] :: [INFO] :: postUser test case completed successfully!!!
9 [03-Nov-2021 08:54:33] :: [GetPostRequest] :: [INFO] :: Inside getUsers test case
10 [03-Nov-2021 08:54:33] :: [GetPostRequest] :: [INFO] :: getUsers test case completed successfully!!!
11 [03-Nov-2021 08:54:37] :: [GetPostRequest] :: [INFO] :: Inside getUsers test case
12 [03-Nov-2021 08:54:37] :: [GetPostRequest] :: [INFO] :: Total Data : 12
13 [03-Nov-2021 08:54:37] :: [GetPostRequest] :: [INFO] :: Per Page Data : 6
14 [03-Nov-2021 08:54:37] :: [GetPostRequest] :: [INFO] :: getUsers test case completed successfully!!!
15 [03-Nov-2021 08:54:37] :: [GetPostRequest] :: [INFO] :: getUsers test case completed successfully!!!
16
17
```