**Testing the CRUD Operations and performance Testing using Jmeter**

DESCRIPTION :

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.

In this document – Performance testing is described

REQUISITES :

●Java 1.8

●Url(https://reqres.in/api/users?page=2)

●JMeter

REQUIREMENTS TO BE MET :

* Check the above web services (GET and POST request of list of users) performance using**JMeter(**
* Do Load Testing.
* Do Stress Testing.
* Do Spike Testing.

**Steps - Load Testing:**

Documentation:

JMeter Load Testing - reqres.in Home Page

This documentation provides guidance on setting up a load test in JMeter to simulate load on the home page of the reqres.in website using a configuration table. The configuration table includes column field names such as start thread count, initial delay, startup time, hold load time, and shut down time.

1. Launch JMeter:

Start JMeter and create a new Test Plan.

2. Add Thread Group:

Right-click on the Test Plan and select "Add > Threads (Users) > Thread Group". This will create a Thread Group element.

3. Configure Thread Group:

In the Thread Group element, specify the following properties based on your load testing requirements:

- Number of Threads (start thread count): Set the number of concurrent users you want to simulate. This determines the load on the system.

- Ramp-Up Period (initial delay): Set the time period in seconds over which the threads/users should be started. It determines the time taken to reach the specified number of threads.

- Loop Count: Set the number of times you want the Thread Group to iterate. For load testing the home page, you can set it to a high number or set it to Forever to run indefinitely.

- Scheduler: You can use the Scheduler options to configure the test duration or specify a specific start and end time.

4. Add HTTP Request Sampler:

Right-click on the Thread Group and select "Add > Sampler > HTTP Request". This will create an HTTP Request Sampler element.

5. Configure HTTP Request Sampler:

In the HTTP Request Sampler element, specify the following properties:

- Server Name or IP: Set the server name or IP address as "reqres.in".

- Path: Set the path as "/".

This will send an HTTP GET request to the reqres.in home page.

6. Add Listeners:

Right-click on the Thread Group and select "Add > Listener" to add listeners for capturing and analyzing the load test results. You can use listeners such as View Results Tree, Summary Report, or Aggregate Report.

7. Configure Load Testing Table:

Create a table with the following column field names and corresponding values:

- Start Thread Count: Specify the number of threads/users to start at the beginning of the test run.

- Initial Delay: Set the initial delay in seconds before starting the load. This can be used to stagger the start of threads.

- Startup Time: Set the time duration in seconds to gradually increase the load up to the desired thread count.

- Hold Load Time: Set the time duration in seconds to maintain the specified thread count.

- Shut Down Time: Set the time duration in seconds to gradually decrease the load down to zero threads.

Use the JMeter CSV Data Set Config element or any other suitable method to read the values from the table.

8. Configure Test Plan:

In the Test Plan, add a Loop Controller and set the loop count based on the number of rows in the load testing table.

9. Configure Thread Group Loop Controller:

Place the Thread Group inside the Loop Controller and set the loop count to the number of rows in the load testing table.

10. Use the Load Testing Table Values:

Within the Loop Controller, use the load testing table values to configure the Thread Group properties dynamically for each iteration.

11. Run the Load Test:

Save the test plan and run the load test by clicking the "Start" button or using the command-line interface.

12. Analyze the Results:

Once the load test is completed, analyze the results using the configured listeners to view response times, throughput, errors, and other performance-related metrics. Response Time Graph , Aggregate Report and Response Times over time listeners are added and the corresponding out are taken in output files

**Output Files:**

* LoadTestAggRep.csv
* LoadTestRTG.csv
* LoadTestRTOT.csv

By following these steps and configuring the load testing table with the specified column field names, start thread count, initial delay, startup time, hold load time, and shut down time, you can perform load testing on the reqres.in home page using JMeter.

**Steps : Stress Testing**

Documentation:

JMeter Stress Testing - reqres.in List Users

This documentation provides guidance on setting up a stress test in JMeter to simulate stress on the "List Users" API endpoint of the reqres.in website. The stress test is configured with the following thread properties: Number of threads, Ramp-up period, Infinite loop, and Thread lifetime duration.

1. Launch JMeter:

Start JMeter and create a new Test Plan.

2. Add Thread Group:

Right-click on the Test Plan and select "Add > Threads (Users) > Thread Group". This will create a Thread Group element.

3. Configure Thread Group:

In the Thread Group element, specify the following properties:

- Number of Threads: Set the number of concurrent users you want to simulate. For stress testing, set it to a high value like 200. This determines the stress on the system.

- Ramp-Up Period: Set the time period in seconds over which the threads/users should be started. For stress testing, set it to a reasonable value like 180 seconds to gradually increase the load.

- Loop Count: Select "Forever" to run the test indefinitely or specify a specific number of loops if needed.

- Scheduler: You can use the Scheduler options to configure the test duration or specify a specific start and end time.

4. Add HTTP Request Sampler:

Right-click on the Thread Group and select "Add > Sampler > HTTP Request". This will create an HTTP Request Sampler element.

5. Configure HTTP Request Sampler:

In the HTTP Request Sampler element, specify the following properties:

- Server Name or IP: Set the server name or IP address as "reqres.in".

- Path: Set the path as "/api/users?page=2".

This will send an HTTP GET request to the reqres.in List Users API endpoint.

6. Add Listeners:

Right-click on the Thread Group and select "Add > Listener" to add listeners for capturing and analyzing the stress test results. You can use listeners such as View Results Tree, Summary Report, or Aggregate Report.

7. Configure Thread Group Properties:

Configure the following properties in the Thread Group to set the stress test parameters:

- Number of Threads: Set it to 200 or the desired number of concurrent users.

- Ramp-Up Period: Set it to 180 seconds or the desired ramp-up duration.

- Loop Count: Select "Forever" to run the test indefinitely.

8. Configure Thread Lifetime Duration:

Right-click on the Thread Group and select "Add > Logic Controller > Throughput Controller". This will create a Throughput Controller element.

Set the Throughput Controller's "Percent Executions" to 100% and "Duration" to 300 seconds or the desired thread lifetime duration.

9. Run the Stress Test:

Save the test plan and run the stress test by clicking the "Start" button or using the command-line interface.

10. Analyze the Results:

Once the stress test is completed, analyze the results using the configured listeners to view response times, throughput, errors, and other performance-related metrics. Summary report, Aggregate report and Response time graph are taken as output file results.

**Output files:**

* StressTestAggGraph.jtl
* StressRespTimeGraph.jtl
* StressTestSummaryReport.csv

By following these steps and configuring the thread properties with the specified number of threads, ramp-up period, infinite loop, and thread lifetime duration, you can perform stress testing on the reqres.in List Users API endpoint using JMeter.

**Spike Testing**

Documentation:

JMeter Spike Testing - reqres.in List Users

This documentation provides guidance on setting up a spike test in JMeter to simulate sudden spikes in load on the "List Users" API endpoint of the reqres.in website using a configuration table. The configuration table includes column field names such as start thread count, initial delay, startup time, hold load time, and shut down time.

1. Launch JMeter:

Start JMeter and create a new Test Plan.

2. Add Thread Group:

Right-click on the Test Plan and select "Add > Threads (Users) > Thread Group". This will create a Thread Group element.

3. Configure Thread Group:

In the Thread Group element, specify the following properties:

- Number of Threads (start thread count): Set the number of concurrent users you want to start at the beginning of the spike.

- Ramp-Up Period (initial delay): Set the initial delay in seconds before starting the spike. This can be used to stagger the start of threads.

- Loop Count: Set the number of times you want the Thread Group to iterate. For spike testing, set it to a high number or set it to Forever to run indefinitely.

- Scheduler: You can use the Scheduler options to configure the test duration or specify a specific start and end time.

4. Add HTTP Request Sampler:

Right-click on the Thread Group and select "Add > Sampler > HTTP Request". This will create an HTTP Request Sampler element.

5. Configure HTTP Request Sampler:

In the HTTP Request Sampler element, specify the following properties:

- Server Name or IP: Set the server name or IP address as "reqres.in".

- Path: Set the path as "/api/users?page=2".

This will send an HTTP GET request to the reqres.in List Users API endpoint.

6. Add Listeners:

Right-click on the Thread Group and select "Add > Listener" to add listeners for capturing and analyzing the spike test results. You can use listeners such as View Results Tree, Summary Report, or Aggregate Report.

7. Configure Spike Testing Table:

Create a table with the following column field names and corresponding values:

- Start Thread Count: Specify the number of threads/users to start at the beginning of the spike.

- Initial Delay: Set the initial delay in seconds before starting the spike. This can be used to stagger the start of threads.

- Startup Time: Set the time duration in seconds to gradually increase the load up to the desired thread count.

- Hold Load Time: Set the time duration in seconds to maintain the specified thread count during the spike.

- Shut Down Time: Set the time duration in seconds to gradually decrease the load down to zero threads.

Use the JMeter CSV Data Set Config element or any other suitable method to read the values from the table.

8. Configure Test Plan:

In the Test Plan, add a Loop Controller and set the loop count based on the number of rows in the spike testing table.

9. Configure Thread Group Loop Controller:

Place the Thread Group inside the Loop Controller and set the loop count to the number of rows in the spike testing table.

10. Use the Spike Testing Table Values:

Within the Loop Controller, use the spike testing table values to configure the Thread Group properties dynamically for each iteration.

11. Run the Spike Test:

Save the test plan and run the spike test by clicking the "Start" button or using the command-line interface.

12. Analyze the Results:

Once the spike test is completed, analyze the results using the configured listeners to view response times, throughput, errors, and other performance-related metrics. Summary Report and Aggregate Report are output files analysed.

**Output files:**

* SpikeTestAggReport.csv
* SpikeTestSummaryReport.csv

By following these steps and configuring the spike testing table with the specified column field names, start thread count, initial delay, startup time, hold load time, and shut down time, you can perform spike testing on the reqres.in List Users API endpoint using JMeter.