Jmeter notes from training

Test plan

chain or serious of steps are going to given— JMeter will execute it

Test plan components:

- Thread group
  - Controllers
  - Samplers
  - Listeners
  - Timers
  - Assertions
  - Configuration elements

**** Flow important ****: each component add as much required no limit.(one or more report, assertions, http)

- Create the **test plan**.
  - Inside the test plan, add a **thread group.**
    - Configure the number of users and ramp-up time in the thread group.
  - Add HTTP **samplers** to the test plan.
  - Add **assertions** to the test plan to validate the results.
  - Add listeners to the test plan for generating reports.

Thread group

○ Number of threads: users count

  ○ Ramp-up period: 1 (every 1 second new user will added until number of threads)

Samplers:

- Add http, ftp, jdbc, … requests
  - HTTP request:
    - Web server for website testing
    - HTTP request for API test

Assertions

- After adding samplers, you can validate with assertions.
  - JSON Assertions
  - Size Assertions
  - Response Assertions
    - Response code

    - Response headers
    - Response data
    - And many more
  - Duration Assertions
  - And many more.

Listeners

- View summary report
    - Latency how much time took for receive first byte
  - Graph report
  - Aggregate report

Generate report:(Jmeter dashboard)
Cli

```
jmeter -n -t Threadgroup.jmx results.jtl -e -o ~/marees/
myoutputPath

.jmx -> test plan // contains one or more thread group
.jtl -> output report
```

Config elements

Http header manager

- For setting up headers(content type, cookie, auth token)

Jmeter Personal notes:

Intro:

- Apache JMeter is an open-source tool used for performance, load, and stress testing of applications.
  - JMeter GUI provides a tree-based(components execute one by one) visual interface for designing, debugging, and executing performance, load, and functional tests.
  - Apache JMeter is an open-source tool used for load, performance, and functional testing of web applications, APIs, and databases. It

simulates real user traffic to assess system behavior under different loads.

⍰ Load Testing - within the requirement (As per the load ex-100 users=100 users)
⍰ Stress Testing - Against the requirement (beyond) ex-100 users = 500 users
⍰ Volume Testing - Transferring a huge amount of data.
⍰ Soak Testing - Transfer huge data within a specific period of a time.

Load Testing:

- **Goal:** Verify system behavior under *expected* user loads (e.g., daily traffic, sales events) to ensure it meets performance SLAs and provides a good user experience.
- **Method:** Gradually increases users/transactions to typical peak levels, focusing on response times, throughput, and resource use.
- **Focus:** Stability and performance within normal operating ranges.

Stress Testing:

- **Goal:** Find the system's breaking point and how it recovers from *extreme* or *unexpected* loads (like DDoS attacks).
- **Method:** Pushes the system far beyond normal capacity, often with a sudden spike or continuous overload, to see where it fails.
- **Focus:** Robustness, maximum capacity, and graceful failure/recovery.

Soak Testing (Endurance Testing)

- **Goal:** Detect subtle performance degradation or resource leaks (memory, disk space) that only appear over extended periods.
- **Method:** Applies a *normal to high* load for a *long duration* (hours/days).
- **Focus:** Long-term stability, memory management, and resource exhaustion.

Key Differences in a Nutshell

- **Load:** Normal Traffic, Typical Duration (Minutes/Hours) – *Is it fast enough for users?*
- **Stress:** Extreme Traffic, Short Duration – *When does it break?*
- **Soak:** Sustained (Avg/High) Traffic, Long Duration – *Does it stay stable over time?*

JMeter Basics:

• - Thread Groups (Users Simulation)

• - Samplers (HTTP, JDBC, FTP, etc.)

• - Listeners (View Results, Graphs)

• - Timers, Assertions, Config Elements

Building a Simple Test Plan

• - Create a Thread Group

• - Add HTTP Sampler to simulate requests

• - Add Listeners to monitor results

• - Run and analyze results

Advanced Test Plans

• - Parameterization with CSV Data Set Config

• - Correlation (Extracting Dynamic Values)

• - Assertions for validation

• - Using Pre-Processors and Post-Processors

Monitoring and Analysis

- - Analyzing Response Time, Throughput, Errors
- - Integrating with Grafana and InfluxDB
- - Generating HTML Test Reports

Real-World Projects - ***need to try this on free time***

- - Load test a REST API
- - Stress test an e-commerce website
- - Database performance testing using JDBC Sampler. —> **need to refer this**

JMeter components detailed:

1. Test Plan
The root of any JMeter test where you define overall configuration like variables, properties, classpaths, etc.
2. Thread Group (Users)
Defines how many users (threads) will run, how often, and how fast. It's the entry point for your execution logic.
3. Samplers
Represent actual requests (HTTP, FTP, JDBC, etc.) sent to a server. Examples include HTTP Request, JDBC Request, SOAP/XML-RPC Request,
4. Logic Controllers
Control the flow of samplers: if, loop, while, switch, transaction, etc. Helps create dynamic execution paths.
5. Config Elements
Provide configuration values to samplers. Examples: HTTP Request Defaults, CSV Data Set Config, Login Config.
6. Pre-Processors
Run before a sampler to set up conditions. Example: modifying request parameters or setting dynamic values.
7. Post-Processors
Run after a sampler to extract or evaluate responses. Examples include Regular Expression Extractor and JSON Extractor.
8. Assertions
Validate responses – check if the server responded correctly using Response Assertion, Duration Assertion, Size Assertion, etc.
9. Timers
Add delay between requests to simulate real-world user behavior (Constant Timer, Gaussian Random Timer, etc.).
10. Listeners
Collect and display results: View Results Tree, Summary Report, Aggregate Report, Graph Results, and more.

JMeter CLI/Headless

```
jmeter —n —t Reqres_TestPlan.jmx —l results.jtl —e —o ./
```

```
html-report
```

- -n: Non-GUI mode
- -t: Path to test plan
- -l: Log results
- -e: Generate HTML report
- -o: Output folder for report

key features of JMeter?
It supports multithreading, protocol testing (HTTP, JDBC, FTP), assertions, data parameterization, CI integration, and visualization with
listeners.

What is a Thread Group in JMeter?
A Thread Group controls the number of users, ramp-up time, and iteration count. It represents how the load is generated in a test plan.

What is a Sampler in JMeter?
Samplers are requests JMeter sends to the server (like HTTP Request, JDBC Request). They simulate actions users perform.

What is a Pre-Processor in JMeter?
It runs before a sampler to set conditions or manipulate data (e.g., setting variables, modifying requests).

What is a Post-Processor?
It processes the sampler's response. Common use: extracting values from response using JSON Extractor or Regular Expression
Extractor.

What is the purpose of the HTTP Header Manager?
It allows setting headers (like Content-Type, Authorization) for HTTP requests, useful for API testing.

What is the difference between Test Plan and Thread Group?
Test Plan is the root container for the test. Thread Group defines users and execution logic inside it.

How do you test a GET API using JMeter?
Add Thread Group → Add HTTP Request with method as GET → Provide path → Add Listeners to verify response.

How do you test a POST API using JMeter?
Use HTTP Request sampler → Select POST → Add body data or parameters → Add headers → Validate with assertions.

What is Throughput in JMeter?

Throughput is the number of requests handled per unit time. Higher throughput = better performance.

What is Latency in JMeter?
Latency is the time taken to get the first byte of the response after the request is sent.

What is the difference between response time and latency?
Response time = time from request to complete response. Latency = time to first byte of the response.

How do you test file upload API in JMeter?
Use HTTP Request → Method POST → Add File Upload tab → Set MIME type and file path.

How do you generate performance test reports?
Run tests with -e -o flags in CLI to generate HTML reports showing response time, errors, and throughput.

Explain the concept of thread groups in JMeter.
A thread group represents a group of virtual users performing a set of operations. You can configure the number of users, ramp-up period, and loop count to define the load for your test.

What is the difference between throughput and hits per second in JMeter?

Answer:

```
Throughput is the amount of data sent to the server (or
received) perunit time, typically measured in requests per
second or transactions per second.

Hits per second refers to the number of HTTP requests made
persecond during the test.
```

What are timers in JMeter? Why are they used?

Answer: Timers in JMeter add delays between requests to simulate real-world user behavior and avoid overwhelming the server with back-to-back requests.
Common timers include:

```
Constant Timer
Gaussian Random Timer
Synchronizing Timer
```

How do you test REST APIs using JMeter?

Answer:

1. Add an HTTP Request Sampler to send API requests.

2. Configure the API endpoint, method (GET/POST), headers, and parameters.
3. Use assertions to validate the response status and content.

How do you perform stress testing in JMeter?
Answer:

1. Increase the number of threads (users) in the Thread Group beyond expected traffic.
2. Configure a high loop count to simulate sustained load.
3. Monitor the server's performance until it breaks or degrades.

Controllers:
Controllers allow you to define the logic of your test. There are two types of controllers:

o Logic Controllers: These control the flow of the requests. For example, you can use a Loop Controller to repeat a request multiple times or an If Controller to define conditional execution.

o Transaction Controllers: Used to group multiple requests as a single transaction for better analysis.

JMeter test elements flow:
If you're testing a login page, you could set up a Thread Group with 100 users, an HTTP Request Sampler to submit the login form, and a Listener to record the response times.

What is the Test plan?
A test plan is a document that stores thread group//samplers//listeners//and multiple operations.
What is the Thread group?
Number of users we must define here lest say from 0-1000 to user need.

What are samplers?
Request, Multiple requests available.(http)

What are Listeners?
output results/report generate for analysis the performance response.

What are Timers?
Types of timers available? Which do we use the most? Timers are the time for the sampler-request to perform when and how at what time. Also, we do have multiple Timers.