# Performance Testing Interview Q&A{ACCOLITE – interview}

**Performance Testing Interview Q&A (As per user question order)**

**1. Correlations**

**Definition:** Correlation in LoadRunner is the process of handling dynamic values (like session IDs or tokens) returned by the server. These values change every session and must be captured and reused.

**Syntax Example:**

web\_reg\_save\_param("sessionID", "LB=SessionID=", "RB=;", LAST);

**Types:**

* Manual Correlation
* Automatic Correlation

**2. Think Time**

**Definition:** Think time is the delay added between actions in a script to simulate real user behavior.

**Syntax:**

lr\_think\_time(5); // 5 seconds delay

**3. Parameterization**

**Definition:** Replacing hardcoded values with variables to simulate different user inputs.

**Syntax for username:**

"username={Username}"

* {Username} is a parameter defined in the parameter list.

**4. Debug Message Syntax**

**Syntax:**

lr\_output\_message("Current Username: %s", lr\_eval\_string("{Username}"));

**Other Debug Functions:**

* lr\_log\_message() – outputs only in Replay log
* lr\_error\_message() – marks as error

**5. Cookies in LoadRunner**

**Syntax:**

web\_add\_cookie("SessionID=abc123; DOMAIN=example.com");

web\_cleanup\_cookies(); // removes all cookies

web\_remove\_cookie("SessionID"); // removes specific cookie

**6. Handling Asynchronous Calls**

**Use:** For AJAX, polling, or long-polling requests.

**Syntax:**

web\_reg\_async\_attributes("ID=asyncCall", "Pattern=Poll", "URL=http://...", LAST);

web\_url("triggerAsync", "URL=http://...", LAST);

web\_stop\_async("ID=asyncCall", LAST);

**7. Custom Header Function**

**Syntax:**

web\_add\_header("Authorization", "Bearer abc123token");

web\_remove\_auto\_header("Header-Name");

**8. Throughput and Response Time**

| **Metric** | **Description** |
| --- | --- |
| Response Time | Time to receive response from server |
| Throughput | Number of requests/data per second |

**Analogy:** Response time = how fast, Throughput = how much.

**9. Performance Bottleneck and Identification**

**Definition:** A bottleneck is a component that limits system performance.

**Common Types:** CPU, Memory, Disk I/O, Network, Database

**How to Identify:**

* Monitor server metrics (CPU, memory)
* Use APM tools (AppDynamics, Dynatrace)
* Correlate metrics with errors or response times

**10. Ramp-Up and Ramp-Down**

**Ramp-Up:** Gradually increasing users  
**Ramp-Down:** Gradually decreasing users

**Example:** 100 users, Ramp-Up: 10 mins → 10 users every minute

**11. Vertical and Horizontal Scaling**

| **Type** | **Description** |
| --- | --- |
| Vertical Scaling | Add more resources to a single machine |
| Horizontal Scaling | Add more machines/nodes to distribute load |

**12. Samplers in JMeter**

**Definition:** Samplers are used to send requests in JMeter.

**Types:**

* HTTP Request
* JDBC Request
* FTP Request
* SOAP/XML-RPC Request
* Debug Sampler
* JSR223 Sampler

**13. Thread Group in JMeter**

**Definition:** Defines the number of users, ramp-up time, and loop count.

**Main Settings:**

* Number of Threads (Users)
* Ramp-Up Period
* Loop Count

**14. Dynamic CSRF Token Handling in JMeter**

**Steps:**

1. Use Regular Expression Extractor to capture token
2. Reuse token using ${csrf\_token} in next request

**Regex Example:**

<input type="hidden" name="csrf\_token" value="(.+?)"/>

**15. Timers in JMeter**

**Purpose:** To add delay between requests.

**Common Timers:**

* Constant Timer
* Gaussian Random Timer
* Uniform Random Timer
* Constant Throughput Timer

**16. HTTP(S) Methods in JMeter**

| **Method** | **Purpose** |
| --- | --- |
| GET | Fetch data |
| POST | Send data (e.g., login) |
| PUT | Replace resource |
| PATCH | Partially update resource |
| DELETE | Remove resource |
| HEAD | Fetch headers only |
| OPTIONS | Fetch allowed methods |

**17. JMeter Functions**

| **Function** | **Use Case** |
| --- | --- |
| ${\_\_time()} | Get current timestamp |
| ${\_\_Random(1,100)} | Get random number |
| ${\_\_UUID()} | Generate unique ID |
| ${\_\_threadNum()} | Get thread number |
| ${\_\_eval()} | Evaluate nested variable |

**18. WebSockets in JMeter**

**Note:** JMeter doesn't support WebSockets natively.

**Solution:** Use "WebSocket Samplers by Peter Doornbosch" plugin.

**Steps:**

1. WebSocket Open Connection
2. WebSocket Send Request
3. WebSocket Read Sampler
4. WebSocket Close Connection

**19. Metrics We Monitor in JMeter**

| **Metric** | **Description** |
| --- | --- |
| Response Time | Time taken for response |
| Throughput | Requests/data per second |
| Latency | Time to first byte |
| Error Rate | % of failed requests |
| Transactions per Sec | Completed requests per second |
| Active Threads | Current virtual users |
| 95th Percentile | 95% requests responded below this |

**Tools:**

* Aggregate Report
* Summary Report
* View Results Tree
* HTML Dashboard