**Why have you applied for this position?**

* I have applied for this job because I believe my experience and skills align closely with the requirements outlined in the job description.
* I had a Working Experience previously in **Target,** so I am well-equipped to handle the challenges of this role.
* I've a good experience in Java coding, where I've developed robust applications and scripts to automate testing processes.
* With over 10+ years of Experience in QA I can Utilize my extensive experience and skills in Java coding, performance testing, and automation.
* Overall, I am eager to utilize my expertise and contribution effectively to the success of the team and the organization.

**JMETER**

1. **What are Samplers and Listeners in JMeter**

In JMeter, Samplers are components responsible for **sending requests** to the server under test, simulating real user actions such as clicking links, filling forms, or making API calls. They generate load on the server by executing the requests defined in test plans.

Listeners, on the other hand, are components used to **collect and display** test results. They capture data generated during test execution, including response times, throughput, and error rates. Listeners provide various visualization options such as tables, graphs, and reports to analyze and interpret test results effectively.

Sure, let's consider an example of using **JMeter** to perform load testing on an e-commerce website. In this scenario, we'll simulate multiple users browsing the website, searching for products, adding items to the cart, and making purchases. We'll use **samplers** to generate requests simulating these user interactions and **listeners** to analyze the performance of the website under load.

2. **How do you run JMeter on non-GUI mode**

To run JMeter in non-GUI mode, you use the command-line interface. You navigate to the "bin" directory of your JMeter installation and execute the JMeter command with the appropriate parameters, including -n for non-GUI mode, -t to specify the test plan file, and -l to specify the results file. This allows you to execute your performance tests efficiently without the graphical interface.

1. Create a test plan using the JMeter GUI and save it with a. jmx extension.
2. Open a terminal or command prompt window.
3. Navigate to the "bin" directory of your JMeter installation using the cd command.
4. Run JMeter in non-GUI mode by executing the JMeter command with the -n parameter for non-GUI mode, -t to specify the test plan file, and -l to specify the results file.
5. Press Enter to execute the test. JMeter will run your test plan in non-GUI mode, and the results will be saved to the specified file.

**Selenium**

**1. How to access disabled elements with selenium**

Accessing disabled elements with Selenium can be challenging because disabled elements are not interactable by default. However, you can still interact with them using **JavaScript executor.**

Example:

Sure, let's say you have a scenario where you need to test a form field that is initially disabled, but you want to input some text into it using Selenium.

In this example:

1. We navigate to a webpage containing a form field (**input** element) with the id **disabled\_input**, which is initially disabled.
2. We find the disabled element using its CSS selector.
3. We check if the element is disabled by using the **get\_attribute ()** method to retrieve the value of the "disabled" attribute.
4. If the element is disabled, we use JavaScript executor to remove the "disabled" attribute from the element.
5. After enabling the element, we can interact with it as usual by using the **send\_keys ()** method to input text into the field.
6. Finally, we close the WebDriver.

This approach allows us to interact with disabled elements in Selenium by programmatically enabling them using JavaScript executor. However, it's important to use this technique judiciously and consider the implications for your testing scenario.

**JAVA**

**1.** What is Method hiding and Variable hiding?

2. Difference between HashMap and Hashtable?

3. How to create a class Immutable?

4. Why String is Immutable?

5. What is Singleton and How do you break the Singleton Class?

6. What are the most used Java8 features in your project?

**API**

**1. Status Codes?**

* **429 -** too many requests in given amount of time ("rate limiting")
* **409** – Conflict

2. Difference between 401 and 403?

3. What is Bearer Token and give me a Realtime Example?

Ans: Consider a scenario where you log in to a social media platform like Facebook or Twitter:

1. **Authentication**: After entering your username and password, the server verifies your credentials and generates a bearer token to represent your authenticated session.
2. **Bearer Token Issuance**: Upon successful authentication, the server issues a bearer token, which is then stored securely on your device (e.g., in local storage, session storage, or cookies).
3. **Accessing Protected Resources**: Whenever you interact with the platform, such as posting a tweet or viewing your timeline, your device automatically includes the bearer token in the request headers.
4. **Server Validation**: The server receives the request and validates the bearer token included in the request headers.
5. **Granting Access**: If the bearer token is valid and has not expired, the server grants access to the requested resource (e.g., posting a tweet, accessing timeline).
6. **Token Expiry and Renewal**: Bearer tokens typically have an expiration time. Once expired, you need to re-authenticate to obtain a new bearer token.

In this example, the bearer token acts as proof of your authentication and allows you to access your account and perform actions on the social media platform without the need to re-enter your username and password with each request.

**JAVA Spring Boot**

1. **How will you check the Health of an Application?**

Spring Actuator is a feature provided by the Spring Boot framework that allows developers to monitor and manage their applications easily. It provides various built-in endpoints and tools to gather information about the application's health, metrics, configuration, environment, and more. In simple terms, Spring Actuator helps developers to understand how their application is performing and diagnose any issues by exposing useful information via HTTP endpoints.

Top of Form

Bottom of Form