techSmartHire

Java SDET - Selenium/API Testing Assessment

Selenium Questions

Question 1: Single Choice

In a Selenium test, you need to wait for an element to become visible and clickable before interaction. The page has dynamic content that loads at different speeds. Which wait strategy is MOST appropriate?

1. Use Thread.sleep(5000) to ensure the element is ready
2. Use driver.manage().timeouts().implicitlyWait() with a 10-second timeout
3. Use WebDriverWait with ExpectedConditions.elementToBeClickable()
4. Use driver.findElement() repeatedly in a loop until the element is found

Correct Answer: Option 3

Explanation: WebDriverWait with ExpectedConditions.elementToBeClickable() is the most reliable approach as it waits for the element to be both visible and enabled, polling at regular intervals until the condition is met or timeout occurs. This is more efficient than fixed waits and more specific than implicit waits.

techSmartHire

Question 2: Single Choice

Question: A Selenium test successfully locates an element using driver.findElement(By.id("submitBtn")) and stores it in a variable. After clicking a "Refresh" link on the same page, attempting to click the stored element reference throws an exception. What is the root cause?

1. The element ID changed after the page refresh

2. The stored WebElement reference became stale after the DOM was refreshed

3. Selenium cannot interact with elements after a page refresh without restarting the browser

4. The implicit wait timeout expired before the element could be clicked

Correct Answer: Option 2

Explanation: When the page is refreshed, the DOM is rebuilt, making the previously stored WebElement reference invalid. This results in a StaleElementReferenceException. The solution is to re-locate the element after the refresh.

techSmartHire

Question 3: Multi-Select

Question: Which of the following are considered GOOD practices when designing a Selenium automation framework? (Select all that apply)

1. Centralizing element locators in Page Object classes rather than embedding them in test methods
2. Using explicit waits with specific expected conditions instead of fixed Thread.sleep() calls
3. Creating test dependencies where Test B requires Test A to execute first to set up data
4. Implementing a base class with common setup and teardown methods for browser management
5. Hardcoding environment URLs directly in test classes for easier maintenance

Correct Answer: Option 1,2, 4

- Option 1: ✓ Page Object Model centralizes locators, reducing maintenance overhead
- Option 2: ✓ Explicit waits are more reliable and efficient than fixed sleeps
- Option 4: ✓ Base classes promote code reuse and consistent browser lifecycle management

techSmartHire

Question 4: Fill in the blanks

Question: In Selenium WebDriver, to switch the driver's focus from the main page to a newly opened browser window or tab, you must first retrieve all window handles and then use the _____

_____

method with the target window handle as a parameter.

```
switchTo().window() or driver.switchTo().window()
```

Correct Answer Explanation

The switchTo().window(windowHandle) method is used to transfer WebDriver's focus to a different browser window or tab, enabling interaction with elements in that context.

techSmartHire

Question 5: Coding Question

Write a program to reverse the given String

java

```
      public class ReverseString {
         public static void main(String[] args) {
           String str = "Selenium";
           String reversed = new
StringBuilder(str).reverse().toString();
           System.out.println(reversed);

            }
     }
```

Or without StringBuilder

java

```
        public class ReverseString {
          public static String reverse (String str) {
            String reversed = "";
            for(int i = str.length() - 1; i >= 0; i--) {
            reversed += str.charAt(i);
        }
        return reversed;
 }
}
```

techSmartHire

Question 6: Single Choice

Question: In a Selenium automation framework, you want to ensure that the browser closes even if a test fails with an exception. Which code structure guarantees this cleanup?

1. Using a try-catch block with driver.quit() in the catch block

2. Using a try-finally block with driver.quit() in the finally block

3. Using a try-catch block with driver.quit() after the catch block

4. Calling driver.quit() at the start of each test method

Correct Answer: Option 2

Explanation: The finally block executes regardless of whether an exception is thrown or caught, making it the ideal place for cleanup code. This ensures driver.quit() runs even when tests fail unexpectedly.

techSmartHire

Question 7: Single Choice

Question: What is the output of the following code?

```
String str1 = "Selenium";
String str2 = str1.concat(" WebDriver");
String str3 = str1;
System.out.println(str1);
System.out.println(str2);
System.out.println(str1 == str3);
```

1. Selenium WebDriver, Selenium WebDriver, false

2. Selenium, Selenium WebDriver, true

3. Selenium WebDriver, Selenium WebDriver, true

4. Compilation error

Correct Answer: Option 2

Explanation: Strings in Java are immutable. The concat() method returns a new String object without modifying the original. str1 remains "Selenium", str2 becomes "Selenium WebDriver", and str1 == str3 is true because both reference the same String object in memory.

techSmartHire

Question 8: Single Choice

Question: During API automation testing, a POST request to create a new user resource returns a status code of 201. What does this indicate?

1. The request was successful, but no content is returned
2. The resource was successfully created
3. The request is being processed asynchronously
4. The resource already exists and was not modified

Correct Answer : Option 2

Explanation: HTTP status code 201 (Created) indicates that the request was successful and a new resource has been created as a result. It's typically used in response to POST requests that create new entities.

techSmartHire

Question 8: Single Choice

Question: When validating a JSON response from an API, you need to verify that a nested field data.user.email exists and contains a valid email format. Which approach is MOST maintainable in a RestAssured test?

1. Parse the response as a String and use .contains("@") to validate email

2. Use JsonPath to extract data.user.email and validate using a regex pattern matcher

3. Convert the entire response to a Java object using Gson and access fields directly

4. Store the response in a file and validate manually after test execution

Correct Answer: Option 2

Explanation: Using JsonPath to extract nested fields provides a clean, maintainable way to validate specific response attributes. Combining it with regex pattern matching allows proper validation of email format without over-engineering the solution. This approach is more flexible than full object deserialization for targeted validations.

techSmartHire

Guidelines & Learning Resources

Prepare yourself to success

⚠ Important Exam Guidelines

Sit in a **quiet place** with a plain background and **good lighting** to ensure a **professional environment**

Ensure **stable internet connection** throughout the exam to avoid disruptions

Keep your **camera ON** at all times during the assessment for proctoring purposes

Do **NOT switch tabs or windows** during the exam - this may result in automatic disqualification

🎓 Learning Resources

Need guidance to clear the assessment with a great score? Check out the systematic learning resources below to learn from scratch.

https://courses.rahulshettyacademy.com/p/java-become-full-stack-qa-automation-expert

☀ All the Best! ☀
You've prepared well, and we believe in your abilities!
Stay calm, focused, and confident during the assessment.
Success is just around the corner!