Web Automation Testing Interview Prep: Essential Questions & Answers

1. Explain Automation Framework

An automation framework is a structured set of guidelines or rules used to create and design test cases for automated testing. It helps to improve the efficiency, effectiveness, and maintainability of tests by providing a systematic approach. Frameworks typically include elements such as test data management, reusable test scripts, reporting, and logging.

2. What is Object Repository?

An Object Repository is a centralized storage location for all the objects (elements) that are used in test scripts. It provides a way to manage and maintain these objects separately from the test scripts, making the scripts more readable and maintainable.

3. Difference Between Page Object and Page Factory

Page Object: A design pattern where each web page is represented by a separate class with methods to interact with the elements on that page. It helps to separate test logic from page-specific operations.

Page Factory: A Selenium feature used in conjunction with the Page Object pattern. It simplifies the initialization of page elements using annotations like @FindBy, making the code more concise and reducing boilerplate code.

4. Types of Automation Frameworks and Their Uses

Data-Driven Framework: Uses external data sources to drive test cases, enabling the execution of the same test with multiple data sets.

Keyword-Driven Framework: Uses keywords to define the actions to be performed, allowing non-technical users to create test cases.

Hybrid Framework: Combines multiple frameworks (e.g., Data-Driven and Keyword-Driven) to leverage their advantages.

Behavior-Driven Development (BDD): Focuses on the behavior of the application using natural language constructs (e.g., Gherkin) for creating test cases.

5. Where Have You Used OOPS Concepts in Your Automation Framework?

Encapsulation: Used to bundle methods and variables related to a particular page or component in a class (e.g., Page Object Model).

Inheritance: Allows sharing common functionalities across different classes by extending a base class.

Polymorphism: Used to define methods in a base class that can be overridden in derived classes for different implementations.

6. Where Have You Used Encapsulation and Abstraction?

Encapsulation: Implemented in Page Object Model by keeping web elements and methods to interact with them within the page class.

Abstraction: Abstract classes or interfaces define common behaviors that are implemented in concrete classes, hiding complex implementations from the user.

7. Difference Between TDD and BDD

Test-Driven Development (TDD): Focuses on writing tests before writing the actual code. The process involves writing a test case, making it pass by writing code, and then refactoring.

Behavior-Driven Development (BDD): Extends TDD by writing tests in natural language that describe the behavior of the application from the user's perspective. It emphasizes collaboration between developers, testers, and business stakeholders.

8. What Are Hooks in BDD?

Hooks are special methods in BDD frameworks (like Cucumber) that allow code to be executed at specific points in the test lifecycle, such as before or after a scenario. Examples include @Before and @After hooks.

9. Explain BDD Structure

BDD structure typically includes:

Features: A description of a feature of the application.

Scenarios: Specific examples or use cases within the feature.

Given-When-Then: A format for writing scenarios, where Given describes the initial context, When describes the action, and Then describes the expected outcome.

10. Difference Between Selenium 3 and Selenium 4

Selenium 4: Introduces a new W3C WebDriver standard for better compatibility, enhanced support for modern browser features, and improved debugging capabilities. Selenium 4 also includes a new Grid for easier management of test infrastructure.

Selenium 3: Used the JSON Wire Protocol and lacked some of the advanced features and improvements available in Selenium 4.

11. Difference Between W3C Protocol and JSON Wire Protocol

W3C Protocol: The standard protocol used by Selenium 4 to communicate with browsers, offering better compatibility and consistency across different browsers.

JSON Wire Protocol: The protocol used by Selenium 3, which has been deprecated in favor of the W3C WebDriver standard.

12. Latest Version of Maven, Selenium, and Java

Maven: 4.0.0 (latest stable release as of August 2024)

Selenium: 4.9.0 (latest stable release as of August 2024)

Java: 21 (latest long-term support version as of August 2024)

13. Types of Loggers

Console Logger: Outputs logs to the console.

File Logger: Writes logs to a file.

Database Logger: Stores logs in a database.

Remote Logger: Sends logs to a remote server.

14. Types of Reports

HTML Reports: Detailed and formatted reports viewable in a web browser.

XML Reports: Structured data reports often used for further processing.

JUnit Reports: XML reports compatible with JUnit results.

Allure Reports: Rich, interactive reports with detailed test results and analytics.

15. TestNG Annotations and Their Execution Sequence

@BeforeSuite

@BeforeTest

@BeforeClass

@BeforeMethod

@Test

@AfterMethod

@AfterClass

@AfterTest

@AfterSuite

16. TestNG Listeners? Which One Will Execute First?

Listeners: ITestListener, ITestNGMethodListener, ITestFailureListener, etc.

Execution Order: ITestListener methods such as onTestStart() execute first before the test methods themselves.

17. Return Type of TestNG Data Provider?

The return type of a TestNG Data Provider is Object[][].

18. TestNG Parameter Use

Parameters can be passed to test methods using @Parameters annotation in TestNG XML files or method-level annotations.

19. Use of POM File

The pom.xml file in Maven projects specifies project dependencies, plugins, and build configurations.

20. Which Maven Dependency Do We Add for Automation?

For Selenium WebDriver, add the dependency:

```
  <dependency>

    <groupId>org.seleniumhq.selenium</groupId>

    <artifactId>selenium-java</artifactId>

    <version>4.9.0</version>

</dependency>
```

21. Difference Between Maven and Gradle

Maven: Uses XML for configuration (pom.xml), follows a convention-over-configuration approach, and has a more rigid lifecycle.

Gradle: Uses Groovy or Kotlin DSL for configuration (build.gradle), offers more flexibility and incremental builds, and supports multiple languages and build tools.

22. How to Take a Screenshot?

Example in Selenium WebDriver (Java)

```
File screenshot = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);

FileUtils.copyFile(screenshot, new File("screenshot.png"));
```

23. Use of Properties? Write Code to Extract Data from .properties File.

Purpose: Store configuration values externally from code for easy updates.

```
Properties prop = new Properties();

FileInputStream input = new FileInputStream("config.properties");

prop.load(input);

String value = prop.getProperty("key");
```

24. Types of Waits and Their Use? When to Use Which Wait?

Implicit Wait: Sets a default waiting time between each test step. Use it when you want a global wait for all elements.

Explicit Wait: Waits for a specific condition before proceeding. Use it for elements that may load at different times.

Fluent Wait: Allows you to define the frequency with which the condition is checked. Use it for complex conditions or scenarios where the wait time varies.

25. What Is the Use of pollingEvery() in Fluent Wait?

pollingEvery(): Defines how frequently the Fluent Wait should check the condition. It helps in reducing CPU usage by adjusting the polling interval.

26. Types of Element Locators

ID

Name

Class Name

Tag Name

Link Text

Partial Link Text

XPath

CSS Selector

27. Between XPath and CSS Selector, Which One Is Faster?

CSS Selector: Generally faster than XPath because it is optimized for performance and does not traverse the DOM.

28. Difference Between XPath and CSS Selector

XPath: Can traverse both up and down the DOM, supports complex queries, and can select elements based on attributes and text.

CSS Selector: Only traverses down the DOM, generally faster, and supports attribute-based selection but has limited capabilities compared to XPath.

29. Locate Element in Amazon Website [using XPath with ::parent or ::ancestor]

Example XPath

//div[@class='s-main-slot']//span[text()='Laptop']//ancestor::div[@class='s-result-item']

30. Which Scenarios Can't Be Automated?

Dynamic Content: Highly dynamic or interactive content that changes frequently.

Visual or Subjective Validation: Scenarios requiring human judgment or visual inspection.

Complex Workflow: Complex end-to-end workflows requiring real-time human interaction.

31. How to Start Automation?

Define Scope: Identify what needs to be automated.

Select Tools: Choose appropriate tools and frameworks (e.g., Selenium, TestNG).

Develop Test Scripts: Write and organize test scripts.

Set Up CI/CD: Integrate with CI/CD tools for continuous testing.

Execute and Maintain: Run tests, analyze results, and maintain the automation suite.

32. Common Exceptions Encountered in Automation? How Do You Deal with Them?

NoSuchElementException: Element not found. Use explicit waits to handle.

TimeoutException: Operation timed out. Increase wait times or optimize conditions.

StaleElementReferenceException: Element is no longer attached to the DOM. Refresh the element or re-fetch it.

33. Interfaces and Classes Used in Selenium Automation

WebDriver Interface: Main interface for browser interaction.

WebElement Interface: Represents HTML elements.

TakesScreenshot Interface: For capturing screenshots.

By Class: Used for locating elements.

34. Is WebDriver an Interface or a Class?

WebDriver is an interface.

35. Which Interfaces and Classes Do You Use to Perform Click, Send Keys, Scroll, Swipe, and Drag and Drop Actions?

Click: WebElement.click()

Send Keys: WebElement.sendKeys()

Scroll: JavascriptExecutor.executeScript()

Swipe: TouchActions or Actions class

Drag and Drop: Actions.dragAndDrop()

36. Code to Select Value from Drop-Down

Example:

```
WebElement dropdown = driver.findElement(By.id("dropdownId"));

Select select = new Select(dropdown);

select.selectByVisibleText("Option Text");
```

37. Use of Jenkins and How to Execute Test Suite After Every 4 Hours Using Jenkins

Use of Jenkins: Jenkins is a continuous integration/continuous deployment (CI/CD) tool used for automating the build, test, and deployment processes.

Schedule Test Suite Execution:

Configure a Jenkins job.

In the job configuration, under "Build Triggers", select "Build periodically".

Use the cron syntax to schedule the job every 4 hours

eg:

```
H */4 * * *
```