Top 50 Automation Testing Interview Q&A

1. What is Selenium and its components?

Selenium is an open-source suite for automating web browsers. Components include Selenium IDE, RC, WebDriver, and Grid.

2. How does WebDriver differ from RC?

WebDriver directly communicates with the browser, while RC uses a server and JavaScript injection.

3. How to handle dynamic elements?

Use XPath functions like contains(), starts-with(), or use dynamic CSS selectors.

4. How to handle dropdowns in Selenium?

Use the Select class. Example: new Select(element).selectByVisibleText("Option");

5. Difference between findElement() vs findElements()

findElement() returns the first match; findElements() returns a list of matching elements.

6. How to handle multiple windows?

Use getWindowHandles() and switchTo().window(handle) to switch between windows.

7. How to capture screenshots?

Use TakesScreenshot interface: ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);

8. Implicit vs Explicit wait - with real examples

Implicit wait applies globally; Explicit wait waits for specific conditions using WebDriverWait.

9. What are Locators in Selenium?

Locators identify elements. Types: ID, Name, Class, TagName, XPath, CSSSelector, LinkText.

10. How to handle file uploads?

Use sendKeys() with file path on <input type='file'> elements.

11. What is Page Object Model?

POM is a design pattern where each page of an app has a corresponding Java class representing it.

12. Difference: Data-driven vs Keyword-driven framework?

Data-driven uses test data from external files. Keyword-driven uses keywords to represent actions.

13. Why Java for automation?

Java is widely supported, has rich libraries, is platform-independent, and integrates well with Selenium.

14. What are TestNG annotations?

TestNG annotations like @Test, @BeforeMethod, @AfterMethod control test execution.

15. Difference between assert and verify?

assert stops execution on failure; verify logs the failure but continues test.

16. How to prioritize test cases in TestNG?

Use priority attribute in @Test annotation: @Test(priority=1)

17. What is a POJO class?

Plain Old Java Object: simple class with private fields, public getters/setters, and a no-arg constructor.

18. What is BDD?

Behavior Driven Development: writing tests in natural language (Gherkin) using tools like Cucumber.

19. Tags, Hooks, Step Definitions in Cucumber?

Tags filter tests. Hooks run before/after scenarios. Step Definitions map steps to Java code.

20. Explain feature file format with example

Feature files use Given-When-Then steps. Example:

Given user logs in

When clicks profile

Then sees profile page.

21. What is Scenario Outline in BDD?

It allows running the same scenario with multiple sets of data using Examples.

22. How to integrate TestNG with Maven?

Add TestNG dependency in pom.xml and run using mvn test command.

23. What is REST API?

REST API follows REST principles using HTTP. CRUD operations map to GET, POST, PUT, DELETE.

24. Difference between PUT and PATCH?

PUT replaces entire resource. PATCH updates partial data.

25. Status Codes: 200 vs 201 vs 204

200: OK, 201: Created, 204: No Content (success but no response body).

26. What is Postman used for?

Postman is a GUI tool to test APIs, automate requests, and validate responses.

27. How to write REST Assured scripts?

Use given().when().get().then().assertThat().statusCode(200); syntax.

28. How to validate response body and headers?

Use body() for content checks and header() for headers validation in Rest Assured.

29. What are path vs query parameters?

Path params are part of URL (/user/{id}); query params come after ? (e.g., ?id=5).

30. Authentication in REST APIs (Bearer, Basic)

Bearer: uses token in header. Basic: base64 encoded username:password in header.

31. What is a testing framework?

Framework defines structure for tests, reports, data handling, and reusable logic.

32. Structure of a Maven-based automation project?

Typical folders: src/main/java, src/test/java, pom.xml, testng.xml, resources.

33. Explain PageFactory in Selenium

PageFactory is used to initialize web elements with @FindBy annotations.

34. What is Singleton design pattern?

Ensures only one instance of a class exists, useful for WebDriver initialization.

35. What is Factory design pattern in Java?

Creates objects without exposing creation logic. Useful for browser or driver instantiation.

36. What are utility/helper classes?

Reusable methods like waits, date functions, or common validations stored in one class.

37. Folder structure of a real-world framework?

Includes test cases, pages, utils, configs, resources, and logs/reports folders.

38. Logging & reporting mechanisms used?

Log4j, ExtentReports, Allure used to log steps and generate reports.

39. What is Jenkins? How to schedule a job?

Jenkins is an automation server to run tests. Use 'Build periodically' with cron syntax.

40. What is a Pipeline in Jenkins?

Pipeline defines stages and steps for CI/CD. Written in Jenkinsfile using Groovy.

41. How to configure Maven in Jenkins?

Install Maven plugin, configure MAVEN_HOME, and use pom.xml to trigger build.

42. Explain Git commands used daily in QA

Common: git clone, pull, add, commit, push, checkout, branch.

43. How to resolve Git merge conflicts?

Use git status to find conflicts, edit files manually, then commit resolved changes.

44. What is Webhook?

Webhook notifies systems (like Jenkins) on GitHub events like push or PRs.

45. Scenario: Element is visible but click fails

Use JavaScriptExecutor to click or check if another element overlays it.

46. Scenario: API returns 500 error, how to debug?

Check request payload, headers, and try the same in Postman or review backend logs.

47. Scenario: Multiple builds failing randomly

Check for flaky tests, timing issues, stale elements, or environment problems.

48. How to handle flaky tests?

Use explicit waits, avoid hardcoded sleeps, retry mechanism, isolate unstable tests.

49. How do you perform cross-browser testing?

Use Selenium Grid, BrowserStack, or cloud platforms to test across browsers.

50. How do you approach writing test cases for a login page?

Include valid, invalid, blank input, UI validations, and security checks.