

SDET Interview Panel – 15 Must-Know SDET Interview Questions

Curated from real panel interviews — with sample answers and practical code examples.

1. Can you walk me through your current Automation Framework design?

What it tests: Tests understanding of architecture, patterns, scaling.

Sample Answer:

My current framework is a modular hybrid framework — combining UI automation using Selenium + TestNG, and API automation using REST Assured. It follows the Page Object Model (POM) design pattern for maintainability. The framework is fully integrated into the CI/CD pipeline (Jenkins), supports parallel execution via ThreadLocal WebDriver, and generates detailed Allure reports. We also follow SOLID principles to ensure the framework is scalable and easy to extend.

2. How is your framework integrated into the CI/CD pipeline?

What it tests: Tests CI/CD knowledge and automation maturity.

Sample Answer:

Our automation suite is integrated into Jenkins. Test execution is triggered automatically on every code push via the pipeline. We run tests in parallel on Selenium Grid and publish Allure reports post-execution. Critical tests run on every build, while full regression runs nightly.

3. How do you handle flaky tests in your current framework?

What it tests: Tests debugging skills and quality mindset.

Sample Answer:

I first analyze the root cause of flakiness — is it the application, environment, or test script? I use explicit waits, improve locator strategies, and stabilize test data. I also tag flaky tests so they're monitored separately, and work with devs to fix underlying issues.

4. How do you prioritize what to automate vs what not to automate?

What it tests: Tests strategic thinking and automation planning.

Sample Answer:

I prioritize based on risk, business impact, frequency of use, and stability of the feature. Critical business flows and high-risk areas come first. I avoid automating highly volatile features or one-time use cases where ROI is low.

5. Can you share an example where you debugged a complex automation failure?

What it tests: Tests real-world debugging and problem-solving.

Sample Answer:

We once had random failures in CI — test cases were timing out inconsistently. After analysis, I found session timeouts due to a recent server config change. I collaborated with DevOps to adjust settings, implemented retry logic, and stabilized the tests.

6. Can you write a simple test scenario using Page Object Model?

What it tests: Tests practical framework skills.

Sample Answer:

In Page Object Model, I create a separate class for each page, containing the locators and methods to interact with that page. My test classes then use these page classes to perform actions and verifications — ensuring clean separation of concerns and better maintainability.

```
// LoginPage.java
public class LoginPage {
    WebDriver driver;

    public LoginPage(WebDriver driver) {
        this.driver = driver;
    }

    By username = By.id("username");
    By password = By.id("password");
    By loginButton = By.id("login");

    public void enterUsername(String user) {
        driver.findElement(username).sendKeys(user);
    }

    public void enterPassword(String pass) {
        driver.findElement(password).sendKeys(pass);
    }

    public void clickLogin() {
        driver.findElement(loginButton).click();
    }
}
```

7. How would you design a reusable test utility for API testing?

What it tests: Tests design thinking and API automation maturity.

Sample Answer:

I would build a utility layer using REST Assured, encapsulating common methods for GET, POST, PUT, DELETE. I would parameterize endpoints, support dynamic headers/auth, and make it easy to chain requests. This helps reduce duplication and improve maintainability.

8. How do you ensure your test code is maintainable and scalable?

What it tests: Tests coding discipline and architecture mindset.

Sample Answer:

I follow SOLID principles, apply design patterns (like POM), and modularize utilities. I enforce code reviews, ensure proper naming, and keep tests independent. I also maintain good documentation and version control for stability.

9. How do you collaborate with DevOps, Developers, and Manual QA in Agile teams?

What it tests: Tests communication and collaboration in Agile.

Sample Answer:

I actively participate in sprint ceremonies, work closely with Devs to identify test hooks, and align with QA to avoid duplication. I collaborate with DevOps on CI/CD optimization. Regular sync-ups ensure that automation aligns with sprint goals.

10. How do you measure the effectiveness of your automation suite?

What it tests: Tests metrics awareness and reporting mindset.

Sample Answer:

I track pass rate trends, failure rate, test coverage, execution time, and defect leakage post-release. We also review flaky test rates. These metrics are reported via dashboards and discussed with the team to drive continuous improvement.

11. How do you manage test data across multiple environments?

What it tests: Tests data management skills and environment readiness.

Sample Answer:

I use a combination of static datasets for consistent validation, dynamic data generators for robustness, and environment-specific configs. For sensitive cases, I leverage mocks/stubs. I ensure data isolation across parallel runs to avoid conflicts.

12. How would you handle test data parametrization in your framework?

What it tests: Tests understanding of Data-Driven Testing.

Sample Answer:

I use TestNG's DataProvider to pass data to test methods. For external data, I use Excel/CSV/JSON readers, keeping data separate from test logic. This allows easy scalability and better maintainability of test cases.

```
@DataProvider(name = "loginData")
public Object[][] loginData() {
    return new Object[][] {
        {"user1", "pass1"},
        {"user2", "pass2"}
    };
}

@Test(dataProvider = "loginData")
public void testLogin(String username, String password) {
    LoginPage loginPage = new LoginPage(driver);
    loginPage.enterUsername(username);
    loginPage.enterPassword(password);
    loginPage.clickLogin();
    Assert.assertTrue(driver.getCurrentUrl().contains("dashboard"));
}
```

13. How do you manage WebDriver instance in parallel execution?

What it tests: Tests understanding of Thread Safety and Parallel Execution.

Sample Answer:

I use ThreadLocal WebDriver to ensure that each test thread gets its own WebDriver instance. This prevents conflicts during parallel execution in TestNG or CI/CD pipeline.

```
public class DriverFactory {
    private static ThreadLocal driver = new ThreadLocal<>();

    public static WebDriver getDriver() {
        return driver.get();
    }

    public static void setDriver(WebDriver driverInstance) {
        driver.set(driverInstance);
    }

    public static void quitDriver() {
        driver.get().quit();
        driver.remove();
    }
}
```

14. How do you implement reusable waits in your framework?

What it tests: Tests handling of dynamic elements and robustness.

Sample Answer:

I create a WaitUtils class with reusable methods using WebDriverWait or FluentWait to handle dynamic elements. This avoids flaky tests and improves reliability.

```
public class WaitUtils {
    WebDriver driver;

    public WaitUtils(WebDriver driver) {
        this.driver = driver;
    }

    public WebElement waitForElement(By locator, int timeoutInSeconds) {
        WebDriverWait wait = new WebDriverWait(driver,
        Duration.ofSeconds(timeoutInSeconds));
        return
        wait.until(ExpectedConditions.visibilityOfElementLocated(locator));
    }
}
```

15. How do you validate API responses effectively?

What it tests: Tests API testing skills and good assertions.

Sample Answer:

I use REST Assured with JSONPath or Hamcrest matchers to validate response status codes, headers, and body content. I ensure that validation is clear and meaningful.

```
given()
    .baseUrl("https://api.example.com")
    .when()
    .get("/users")
    .then()
    .statusCode(200)
    .body("size()", greaterThan(0))
    .body("[0].email", containsString("@example.com"));
```