

Topic 1: Writing basic BDD Cucumber Gherkin scenarios & executing with TestNG

Multiple Choice Question 1

Which of the following best represents a valid Gherkin scenario?

Options:

- A. Scenario: User logs in with valid credentials
Then user is redirected to dashboard
- B. Given user is on login page
When user enters credentials
Then user is redirected to dashboard
- C. Feature: Login functionality
Scenario: Login with valid credentials
Then user is redirected
- D. Scenario: Login
Given user opens browser
Then user logs in

Skill: BDD with Cucumber

Subskill: Gherkin syntax

Competency: Writing valid Gherkin scenarios

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option B follows the correct Gherkin syntax with Given-When-Then steps under an implied scenario.
 - **Incorrect Options Feedback:**
 - o A: Missing Given and When steps.
 - o C: Starts with Feature but lacks Given/When/Then structure.
 - o D: Missing When step and lacks clarity.
-

Multiple Choice Question 2

In a Cucumber-TestNG setup, what is the role of the @CucumberOptions annotation?

Options:

- A. It defines the TestNG suite file
- B. It specifies the location of feature files and glue code
- C. It initializes the WebDriver
- D. It maps Gherkin steps to Java methods

Skill: BDD with Cucumber

Subskill: TestNG integration

Competency: Configuring Cucumber with TestNG

Difficulty Level: Intermediate

Bloom Level: Knowledge

- **Correct Answer Reason:**
Option B is correct because @CucumberOptions is used to configure feature file paths and step definitions.
 - **Incorrect Options Feedback:**
 - o A: TestNG suite files are XML-based, not annotations.
 - o C: WebDriver is initialized in step definitions, not via this annotation.
 - o D: Step mapping is done via annotations like @Given, not @CucumberOptions.
-

Multiple Choice Question 3

Which TestNG annotation is used to execute Cucumber scenarios?

Options:

- A. @Test
- B. @RunWith
- C. @CucumberTest
- D. @CucumberOptions

Skill: BDD with Cucumber

Subskill: TestNG execution

Competency: Executing feature files

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option A is correct because TestNG uses @Test to execute methods, including those that run Cucumber scenarios.
 - **Incorrect Options Feedback:**
 - o B: @RunWith is used in JUnit, not TestNG.
 - o C: No such annotation exists in Cucumber.
 - o D: @CucumberOptions is for configuration, not execution.
-

✓ Topic 2: Data-Driven Testing — TestNG @DataProvider, Excel Integration (Apache POI)

Multiple Choice Question 1

What is the primary purpose of the @DataProvider annotation in TestNG?

Options:

- A. To group multiple test methods
- B. To provide test data to a test method
- C. To execute tests in parallel
- D. To log test results

Skill: TestNG

Subskill: Data-driven testing

Competency: Supplying dynamic data to tests

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option B is correct because @DataProvider allows you to pass multiple sets of data to a test method.
 - **Incorrect Options Feedback:**
 - o A: @Test(groups=...) is used for grouping.
 - o C: Parallel execution is configured separately.
 - o D: Logging is handled by listeners or reporting tools.
-

Multiple Choice Question 2

Which Apache POI class is used to read .xlsx Excel files?

Options:

- A. HSSFWorkbook
- B. XSSFWorkbook
- C. WorkbookFactory
- D. ExcelReader

Skill: Apache POI

Subskill: Excel integration

Competency: Reading Excel files

Difficulty Level: Intermediate

Bloom Level: Knowledge

- **Correct Answer Reason:**
Option B is correct because XSSFWorkbook is designed for .xlsx format files.
 - **Incorrect Options Feedback:**
 - o A: HSSFWorkbook is for .xls files.
 - o C: WorkbookFactory is a utility but not the core reader.
 - o D: Not a valid POI class.
-

Multiple Choice Question 3

How can you link Excel data to a TestNG @DataProvider method?

Options:

- A. Use JDBC to connect to Excel
- B. Use Apache POI to read Excel and return Object[][]
- C. Use Selenium WebDriver to scrape Excel
- D. Use @Parameters annotation with Excel path

Skill: TestNG + Apache POI

Subskill: Data-driven testing

Competency: Integrating external data sources

Difficulty Level: Advanced

Bloom Level: Analysis

- **Correct Answer Reason:**

Option B is correct because Apache POI reads Excel and the data is returned as a 2D Object array for TestNG.

- **Incorrect Options Feedback:**

- o A: JDBC is not typically used for Excel.
 - o C: Selenium is for web automation, not file reading.
 - o D: @Parameters is for static XML-based parameters.
-

Topic 3: Reporting — ExtentReports Integration with TestNG

Multiple Choice Question 1

Which class is used to create an ExtentReports instance?

Options:

- A. ExtentTest
- B. ExtentHtmlReporter
- C. ExtentReports
- D. ReportManager

Skill: Reporting

Subskill: ExtentReports setup

Competency: Initializing report generation

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**

Option C is correct because ExtentReports is the main class used to create and manage reports.

- **Incorrect Options Feedback:**

- o A: ExtentTest is used to log individual test steps.
 - o B: ExtentHtmlReporter configures the output format.
 - o D: Not a valid ExtentReports class.
-

Multiple Choice Question 2

What is the correct sequence to log a test step in ExtentReports?

Options:

- A. createTest → log → flush
- B. log → createTest → flush
- C. flush → createTest → log
- D. createTest → flush → log

Skill: Reporting

Subskill: Logging test steps

Competency: Using ExtentReports API

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option A is correct: you first create a test, log steps, and finally flush the report.
 - **Incorrect Options Feedback:**
 - o B: Logging before creating a test is invalid.
 - o C: Flushing before logging will miss data.
 - o D: Flushing must be the final step.
-

Multiple Choice Question 3

How can ExtentReports be integrated with TestNG lifecycle?

Options:

- A. Using @BeforeSuite and @AfterSuite annotations
- B. Using @Test only
- C. Using @BeforeMethod and @AfterMethod only
- D. Using @DataProvider

Skill: Reporting

Subskill: TestNG integration

Competency: Lifecycle management

Difficulty Level: Advanced

Bloom Level: Analysis

- **Correct Answer Reason:**
Option A is correct because initializing and flushing ExtentReports is best done at the suite level.
 - **Incorrect Options Feedback:**
 - o B: @Test runs the test but doesn't manage lifecycle.
 - o C: These are useful for logging per test, not full report lifecycle.
 - o D: @DataProvider is unrelated to reporting.
-

✓ Topic 4: Postman Collections, Requests, Headers, Parameters

Multiple Choice Question 1

What is the primary purpose of a Postman Collection?

Options:

- A. To store environment variables
- B. To group related API requests
- C. To log test results
- D. To generate documentation automatically

Skill: Postman

Subskill: Collections

Competency: Organizing API workflows

Difficulty Level: Intermediate

Bloom Level: Knowledge

- **Correct Answer Reason:**
Option B is correct because Collections are used to group and manage related API requests.
 - **Incorrect Options Feedback:**
 - o A: Environment variables are stored separately.
 - o C: Logging is done via test scripts or external tools.
 - o D: Documentation is a feature, but not the primary purpose.
-

Multiple Choice Question 2

Which header is commonly required when sending JSON data in a POST request?

Options:

- A. Accept: text/html
- B. Content-Type: application/json
- C. Authorization: Bearer token
- D. Cache-Control: no-cache

Skill: Postman

Subskill: Headers

Competency: Configuring request metadata

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option B is correct because it tells the server the request body is in JSON format.
 - **Incorrect Options Feedback:**
 - o A: Accept header defines expected response type, not request body.
 - o C: Authorization is for access control, not content type.
 - o D: Cache-Control affects caching, not content format.
-

Multiple Choice Question 3

In Postman, where are query parameters added for a GET request?

Options:

- A. In the request body
- B. In the Headers tab
- C. In the Params tab
- D. In the Authorization tab

Skill: Postman

Subskill: Parameters

Competency: Configuring request inputs

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option C is correct because the Params tab allows you to add key-value query parameters.
 - **Incorrect Options Feedback:**
 - o A: GET requests don't use a body.
 - o B: Headers are metadata, not query parameters.
 - o D: Authorization is for access control.
-

✓ Topic 5: Postman Assertions & Writing Test Scripts in Postman

Multiple Choice Question 1

Which Postman function is used to assert that a response status is 200?

Options:

- A. `pm.expect(response.status).to.equal(200)`
- B. `pm.test("Status code is 200", function () { pm.response.to.have.status(200); })`
- C. `pm.assert(response.statusCode === 200)`
- D. `pm.response.statusCode(200)`

Skill: Postman

Subskill: Test scripts

Competency: Writing assertions

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option B is correct because it uses Postman's built-in `pm.test` and assertion syntax.
 - **Incorrect Options Feedback:**
 - o A: Incorrect syntax for Postman's assertion library.
 - o C: `pm.assert` is not a valid Postman function.
 - o D: `statusCode()` is not a valid method.
-

Multiple Choice Question 2

Where are Postman test scripts written?

Options:

- A. In the Authorization tab
- B. In the Pre-request Script tab
- C. In the Tests tab
- D. In the Body tab

Skill: Postman

Subskill: Test scripts

Competency: Locating scripting interface

Difficulty Level: Intermediate

Bloom Level: Knowledge

- **Correct Answer Reason:**
Option C is correct because the Tests tab is specifically designed for writing assertions and validations.
 - **Incorrect Options Feedback:**
 - o A: Authorization tab is for credentials.
 - o B: Pre-request scripts run before the request, not after.
 - o D: Body tab is for request payload.
-

Multiple Choice Question 3

Which Postman object provides access to the response body?

Options:

- A. pm.request.body
- B. pm.response.body
- C. pm.response.json()
- D. Pm.test.body

Skill: Postman

Subskill: Test scripts

Competency: Accessing response data

Difficulty Level: Advanced

Bloom Level: Application

- **Correct Answer Reason:**
Option C is correct because `pm.response.json()` parses the response body as JSON.
 - **Incorrect Options Feedback:**
 - o A: Refers to the request body, not response.
 - o B: body is not a direct property; needs parsing.
 - o D: `pm.test` is for assertions, not data access.
-

✓ Topic 6: Postman Authentication — Basic, Bearer Token, OAuth2

Multiple Choice Question 1

Which header is automatically added when using Bearer Token authentication in Postman?

Options:

- A. Authorization: Basic <token>
- B. Authorization: Bearer <token>
- C. Token: Bearer <token>
- D. Access-Control: Bearer <token>

Skill: Postman

Subskill: Authentication

Competency: Configuring Bearer Token

Difficulty Level: Intermediate

Bloom Level: Knowledge

- **Correct Answer Reason:**
Option B is correct because Bearer Token authentication uses the Authorization header with the "Bearer" scheme.
 - **Incorrect Options Feedback:**
 - o A: Basic auth uses base64-encoded credentials.
 - o C: Token is not a standard header.
 - o D: Access-Control is unrelated to authentication.
-

Multiple Choice Question 2

What is required to perform Basic Authentication in Postman?

Options:

- A. OAuth2 token
- B. Username and password
- C. API key
- D. JSON Web Token

Skill: Postman

Subskill: Authentication

Competency: Using Basic Auth

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option B is correct because Basic Auth requires a username and password encoded in base64.
 - **Incorrect Options Feedback:**
 - o A: OAuth2 is a separate protocol.
 - o C: API keys are used differently.
 - o D: JWTs are typically used with Bearer Token.
-

Multiple Choice Question 3

Which Postman feature supports OAuth2 token retrieval?

Options:

- A. Pre-request Script
- B. Authorization tab
- C. Tests tab
- D. Headers tab

Skill: Postman

Subskill: Authentication

Competency: Managing OAuth2 flow

Difficulty Level: Advanced

Bloom Level: Application

- **Correct Answer Reason:**
Option B is correct because the Authorization tab allows configuration and retrieval of OAuth2 tokens.
 - **Incorrect Options Feedback:**
 - o A: Pre-request scripts can manipulate tokens but not retrieve them.
 - o C: Tests tab is for assertions.
 - o D: Headers tab is for manual entry, not token flow.
-

✓ Topic 7: Postman — Chaining Requests Using Variables & Environments

Multiple Choice Question 1

How can you pass a value from one Postman request to another?

Options:

- A. By using `pm.variables.set()` in the Tests tab
- B. By using `pm.environment.set()` in the Pre-request Script tab
- C. By using `pm.globals.get()` in the Body tab
- D. By using `pm.request.set()` in the Headers tab

Skill: Postman

Subskill: Variable chaining

Competency: Sharing data across requests

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option A is correct because `pm.variables.set()` stores a variable that can be accessed in subsequent requests.
 - **Incorrect Options Feedback:**
 - o B: `pm.environment.set()` is valid but used in Tests tab, not Pre-request.
 - o C: `pm.globals.get()` retrieves but doesn't set values.
 - o D: `pm.request.set()` is not a valid method.
-

Multiple Choice Question 2

Which Postman variable scope has the highest precedence?

Options:

- A. Global
- B. Environment
- C. Collection
- D. Local

Skill: Postman

Subskill: Variable scoping

Competency: Managing variable resolution

Difficulty Level: Advanced

Bloom Level: Analysis

- **Correct Answer Reason:**
Option D is correct because local variables (defined within a request) override all other scopes.
 - **Incorrect Options Feedback:**
 - o A: Global has the lowest precedence.
 - o B: Environment is overridden by local.
 - o C: Collection is overridden by environment and local.
-

Multiple Choice Question 3

What is the correct syntax to reference an environment variable in a request URL?

Options:

- A. {{env.variable}}
- B. {{environment.variable}}
- C. {{variable}}
- D. \$env.variable

Skill: Postman

Subskill: Environment variables

Competency: Referencing variables in requests

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option C is correct because Postman uses double curly braces to reference any variable by name.
- **Incorrect Options Feedback:**
 - o A & B: Incorrect syntax; Postman doesn't use dot notation in variable

references.

o D: \$env is not a valid Postman syntax.

✓ Topic 8: Postman — Parameterization with Environment/Global Variables

Multiple Choice Question 1

Which of the following is a valid use case for global variables in Postman?

Options:

- A. Storing frequently changing API keys
- B. Storing base URLs for different environments
- C. Sharing values across multiple collections
- D. Storing request-specific data

Skill: Postman

Subskill: Global variables

Competency: Managing reusable data

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option C is correct because global variables are accessible across all collections and environments.
- **Incorrect Options Feedback:**
 - o A: API keys should be stored in environment variables.
 - o B: Base URLs vary by environment, not globally.
 - o D: Request-specific data should use local variables.

Multiple Choice Question 2

How do you set an environment variable dynamically in a test script?

Options:

- A. `pm.environment.set("key", value)`
- B. `pm.setEnvironment("key", value)`
- C. `pm.env.set("key", value)`
- D. `pm.set("environment.key", value)`

Skill: Postman

Subskill: Environment variables

Competency: Dynamic variable assignment

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option A is correct because `pm.environment.set()` is the correct method to assign environment variables.
 - **Incorrect Options Feedback:**
 - o B: Deprecated syntax.
 - o C: `pm.env` is not a valid object.
 - o D: Incorrect syntax structure.
-

Multiple Choice Question 3

Which tab in Postman allows you to select an active environment?

Options:

- A. Authorization
- B. Tests
- C. Environment Manager
- D. Environment dropdown in the top-right corner

Skill: Postman

Subskill: Environment selection

Competency: Switching environments

Difficulty Level: Intermediate

Bloom Level: Knowledge

- **Correct Answer Reason:**
Option D is correct because the environment dropdown is where users select the active environment.
 - **Incorrect Options Feedback:**
 - o A: Authorization is for credentials.
 - o B: Tests tab is for assertions.
 - o C: Environment Manager is for creating/editing, not selecting.
-

Topic 9: Automating API Tests Using Java, Selenium & Rest Assured

Multiple Choice Question 1

Which Rest Assured method is used to validate the status code of a response?

Options:

- A. `assertEquals(response.statusCode(), 200)`
- B. `response.statusCode(200)`
- C. `response.then().statusCode(200)`
- D. `expect().statusCode(200)`

Skill: API Automation
Subskill: Rest Assured
Competency: Validating API responses
Difficulty Level: Intermediate
Bloom Level: Application

- **Correct Answer Reason:**
Option C is correct because `then().statusCode()` is the standard Rest Assured syntax for assertions.
 - **Incorrect Options Feedback:**
 - o A: JUnit-style, not Rest Assured.
 - o B: Incorrect method chaining.
 - o D: `expect()` is deprecated.
-

Multiple Choice Question 2

Which Maven dependency is required to use Rest Assured?

Options:

- A. `org.seleniumhq.restassured`
- B. `io.restassured`
- C. `com.restassured.api`
- D. `org.apache.restassured`

Skill: API Automation
Subskill: Java + Rest Assured
Competency: Project setup
Difficulty Level: Intermediate
Bloom Level: Knowledge

- **Correct Answer Reason:**
Option B is correct because `io.restassured` is the official group ID for Rest Assured.
 - **Incorrect Options Feedback:**
 - o A, C, D: These are incorrect or non-existent group IDs.
-

Multiple Choice Question 3

Which of the following best describes the role of `given()` in Rest Assured?

Options:

- A. It defines the expected response
- B. It sets up request specifications
- C. It executes the HTTP call
- D. It validates the response body

Skill: API Automation
Subskill: Rest Assured
Competency: Structuring API calls
Difficulty Level: Intermediate
Bloom Level: Application

- **Correct Answer Reason:**
Option B is correct because `given()` is used to define headers, parameters, and body before making the request.
 - **Incorrect Options Feedback:**
 - o A: `then()` handles expectations.
 - o C: `when()` triggers the request.
 - o D: `then().body()` is used for validation.
-

Topic 10: Appium — Locating Mobile Elements (id, accessibilityId, XPath, className)

Multiple Choice Question 1

Which locator strategy is most efficient and preferred for native mobile apps?

- Options:
- A. XPath
 - B. className
 - C. accessibilityId
 - D. id

Skill: Appium
Subskill: Element location
Competency: Choosing optimal locator strategies
Difficulty Level: Intermediate
Bloom Level: Evaluation

- **Correct Answer Reason:**
Option C is correct because `accessibilityId` is fast, stable, and cross-platform for native apps.
 - **Incorrect Options Feedback:**
 - o A: XPath is brittle and slower.
 - o B: `className` may match multiple elements.
 - o D: `id` is useful but platform-dependent.
-

Multiple Choice Question 2

Which locator strategy is most prone to breaking due to UI changes?

Options:

- A. accessibilityId
- B. id
- C. XPath
- D. name

Skill: Appium

Subskill: Locator reliability

Competency: Identifying fragile locators

Difficulty Level: Intermediate

Bloom Level: Analysis

- **Correct Answer Reason:**
Option C is correct because XPath relies on UI hierarchy, which changes frequently.
 - **Incorrect Options Feedback:**
 - o A: accessibilityId is stable across UI changes.
 - o B: id is tied to resource identifiers.
 - o D: name is deprecated in many contexts.
-

Multiple Choice Question 3

Which locator strategy is best suited for cross-platform mobile testing?

Options:

- A. id
- B. accessibilityId
- C. className
- D. XPath

Skill: Appium

Subskill: Cross-platform testing

Competency: Selecting portable locators

Difficulty Level: Advanced

Bloom Level: Evaluation

- **Correct Answer Reason:**
Option B is correct because accessibilityId works consistently across Android and iOS.
- **Incorrect Options Feedback:**
 - o A: id differs between platforms.
 - o C: className varies by OS.
 - o D: XPath is platform-agnostic but fragile.

✓ Topic 11: Appium Script — Launch App, Find Element, Click, sendKeys

Multiple Choice Question 1

Which Appium method is used to launch the mobile application?

Options:

- A. driver.startApp()
- B. driver.launchApp()
- C. driver.openApp()
- D. driver.runApp()

Skill: Appium

Subskill: App lifecycle

Competency: Launching apps via script

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option B is correct because launchApp() starts the app specified in desired capabilities.
- **Incorrect Options Feedback:**
 - o A: Not a valid Appium method.
 - o C: openApp() is not defined in Appium.
 - o D: runApp() is not a recognized method.

Multiple Choice Question 2

Which method is used to send text input to a mobile element?

Options:

- A. element.sendKeys("text")
- B. element.setText("text")
- C. driver.type("text")
- D. element.input("text")

Skill: Appium

Subskill: Element interaction

Competency: Simulating user input

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option A is correct because sendKeys() is the standard method for inputting text.

- **Incorrect Options Feedback:**
 - o B: `setText()` is not universally supported.
 - o C: `driver.type()` is not valid.
 - o D: `input()` is not a recognized method.
-

Multiple Choice Question 3

Which method is used to simulate a tap on a mobile element?

Options:

- A. `element.click()`
- B. `driver.tap()`
- C. `element.tap()`
- D. `driver.clickElement()`

Skill: Appium

Subskill: Element interaction

Competency: Performing tap actions

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**

Option A is correct because `click()` is the standard method for tapping elements.
 - **Incorrect Options Feedback:**
 - o B: `tap()` is deprecated in favor of `TouchAction`.
 - o C: `element.tap()` is not supported directly.
 - o D: `clickElement()` is not a valid method.
-

Topic 12: Synchronization & Waits in Appium

Multiple Choice Question 1

Which type of wait is preferred for handling dynamic elements in Appium?

Options:

- A. `Thread.sleep()`
- B. Implicit wait
- C. Explicit wait
- D. Static wait

Skill: Appium

Subskill: Synchronization

Competency: Managing dynamic UI timing

Difficulty Level: Intermediate

Bloom Level: Evaluation

- **Correct Answer Reason:**
Option C is correct because explicit waits target specific conditions and elements.
 - **Incorrect Options Feedback:**
 - o A: Thread.sleep() is inefficient and blocks execution.
 - o B: Implicit waits apply globally and may cause delays.
 - o D: Static waits are unreliable.
-

Multiple Choice Question 2

Which Appium class is used to implement explicit waits?

Options:

- A. WebDriverWait
- B. FluentWait
- C. AppiumWait
- D. WaitHandler

Skill: Appium

Subskill: Wait implementation

Competency: Using wait classes

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option A is correct because WebDriverWait is used to define explicit wait conditions.
 - **Incorrect Options Feedback:**
 - o B: FluentWait is valid but more advanced.
 - o C: AppiumWait is not a defined class.
 - o D: WaitHandler is not part of Appium.
-

Multiple Choice Question 3

Which condition is commonly used with explicit waits?

Options:

- A. ExpectedConditions.elementToBeClickable()
- B. driver.waitForElement()
- C. element.isDisplayed()
- D. driver.isElementPresent()

Skill: Appium
Subskill: Wait conditions
Competency: Waiting for element readiness
Difficulty Level: Intermediate
Bloom Level: Application

- **Correct Answer Reason:**
Option A is correct because `elementToBeClickable()` ensures the element is ready for interaction.
 - **Incorrect Options Feedback:**
 - o B: Not a valid Appium method.
 - o C: `isDisplayed()` is a check, not a wait condition.
 - o D: `isElementPresent()` is not a standard method.
-

✓ Topic 13: Appium Gestures — Tap, Long Press, Swipe, Scroll, Drag & Drop

Multiple Choice Question 1

Which Appium class is commonly used to perform gestures like tap and swipe?

Options:

- A. `MobileElement`
- B. `TouchAction`
- C. `GestureController`
- D. `ActionBuilder`

Skill: Appium
Subskill: Gesture automation
Competency: Performing mobile gestures
Difficulty Level: Intermediate
Bloom Level: Application

- **Correct Answer Reason:**
Option B is correct because `TouchAction` provides methods like `tap()`, `press()`, and `moveTo()` for gestures.
 - **Incorrect Options Feedback:**
 - o A: `MobileElement` represents UI elements, not gestures.
 - o C: Not a valid Appium class.
 - o D: `ActionBuilder` is used in Selenium, not Appium.
-

Multiple Choice Question 2

Which method is used to perform a long press gesture in Appium?

Options:

- A. `longPress()`
- B. `press().waitAction().release()`
- C. `tap().hold().release()`
- D. `clickAndHold()`

Skill: Appium

Subskill: Gesture control

Competency: Simulating long press

Difficulty Level: Advanced

Bloom Level: Application

- **Correct Answer Reason:**
Option B is correct because chaining `press()`, `waitAction()`, and `release()` simulates a long press.
 - **Incorrect Options Feedback:**
 - o A: `longPress()` is deprecated or not supported in newer versions.
 - o C: `hold()` is not a valid method.
 - o D: `clickAndHold()` is from Selenium.
-

Multiple Choice Question 3

Which gesture is best suited for navigating through vertically scrollable content?

Options:

- A. Tap
- B. Swipe
- C. Scroll
- D. Drag & Drop

Skill: Appium

Subskill: Gesture selection

Competency: Navigating mobile UI

Difficulty Level: Intermediate

Bloom Level: Evaluation

- **Correct Answer Reason:**
Option C is correct because `scroll()` is designed for navigating through scrollable views.
 - **Incorrect Options Feedback:**
 - o A: Tap is for selection, not navigation.
 - o B: Swipe may work but is less precise.
 - o D: Drag & Drop is for rearranging elements.
-

✓ Topic 14: Appium — Handling Keyboard Interactions & Orientation Changes

Multiple Choice Question 1

Which Appium method hides the on-screen keyboard?

Options:

- A. driver.hideKeyboard()
- B. driver.closeKeyboard()
- C. driver.dismissKeyboard()
- D. driver.keyboardOff()

Skill: Appium

Subskill: Keyboard handling

Competency: Managing mobile input UI

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option A is correct because hideKeyboard() is the standard method to dismiss the keyboard.
- **Incorrect Options Feedback:**
 - o B, C, D: These are not valid Appium methods.

Multiple Choice Question 2

Which method is used to change the screen orientation in Appium?

Options:

- A. driver.setOrientation()
- B. driver.rotate()
- C. driver.changeOrientation()
- D. driver.switchOrientation()

Skill: Appium

Subskill: Orientation control

Competency: Managing device orientation

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option A is correct because setOrientation(ScreenOrientation.LANDSCAPE) or PORTRAIT is the correct usage.
- **Incorrect Options Feedback:**
 - o B: rotate() is deprecated or unsupported.
 - o C & D: Not valid Appium methods.

Multiple Choice Question 3

Which Appium enum is used to specify orientation?

Options:

- A. OrientationType
- B. ScreenOrientation
- C. DeviceOrientation
- D. ViewOrientation

Skill: Appium

Subskill: Orientation handling

Competency: Specifying orientation values

Difficulty Level: Intermediate

Bloom Level: Knowledge

- **Correct Answer Reason:**
Option B is correct because ScreenOrientation provides LANDSCAPE and PORTRAIT values.
- **Incorrect Options Feedback:**
 - o A, C, D: These are not valid Appium enums.

 [Topic 15: Appium — Automating Hybrid Apps \(Switching Context: Native ↔ WebView\)](#)

Multiple Choice Question 1

Which Appium method retrieves available contexts in a hybrid app?

Options:

- A. driver.getContexts()
- B. driver.listContexts()
- C. driver.fetchContexts()
- D. driver.getAvailableViews()

Skill: Appium

Subskill: Hybrid app automation

Competency: Identifying app contexts

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option A is correct because getContexts() returns a list of available contexts like NATIVE_APP and WEBVIEW.

- **Incorrect Options Feedback:**
 - o B & C: Not valid Appium methods.
 - o D: `getAvailableViews()` is not used for context switching.
-

Multiple Choice Question 2

What is the correct method to switch to a WebView context?

Options:

- A. `driver.switchContext("WEBVIEW")`
- B. `driver.setContext("WEBVIEW")`
- C. `driver.context("WEBVIEW")`
- D. `driver.changeContext("WEBVIEW")`

Skill: Appium

Subskill: Context switching

Competency: Navigating hybrid app layers

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**

Option C is correct because `context()` is the method used to switch between `NATIVE_APP` and `WEBVIEW`.
 - **Incorrect Options Feedback:**
 - o A, B, D: These are not valid Appium methods.
-

Multiple Choice Question 3

Why is it necessary to switch context when automating hybrid apps?

Options:

- A. To access device logs
- B. To interact with native elements
- C. To interact with web elements inside WebView
- D. To change screen orientation

Skill: Appium

Subskill: Hybrid app testing

Competency: Understanding context switching

Difficulty Level: Intermediate

Bloom Level: Analysis

- **Correct Answer Reason:**

Option C is correct because WebView elements require switching to the `WEBVIEW` context.

- **Incorrect Options Feedback:**
 - o A: Logs are unrelated to context.
 - o B: Native elements are accessible in NATIVE_APP context.
 - o D: Orientation is handled separately.
-

✓ Topic 16: Debugging Locator Issues with Appium Inspector

Multiple Choice Question 1

What is the primary use of Appium Inspector?

Options:

- A. To record test scripts
- B. To view device logs
- C. To inspect UI elements and their properties
- D. To simulate gestures

Skill: Appium

Subskill: Debugging

Competency: Identifying element locators

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**

Option C is correct because Appium Inspector helps visualize the UI hierarchy and inspect element attributes.
 - **Incorrect Options Feedback:**
 - o A: Recording is not its main function.
 - o B: Logs are accessed via Appium server or ADB.
 - o D: Gestures are simulated via scripts, not Inspector.
-

Multiple Choice Question 2

Which attribute is most commonly used to locate elements in Appium Inspector?

Options:

- A. text
- B. resource-id
- C. accessibilityId
- D. bounds

Skill: Appium

Subskill: Locator strategy

Competency: Choosing reliable attributes

Difficulty Level: Intermediate

Bloom Level: Analysis

- **Correct Answer Reason:**
Option C is correct because accessibilityId is stable and cross-platform, making it ideal for automation.
 - **Incorrect Options Feedback:**
 - o A: text can change frequently.
 - o B: resource-id is Android-specific.
 - o D: bounds are positional, not semantic.
-

Multiple Choice Question 3

Which of the following helps resolve locator flakiness in Appium?

Options:

- A. Using absolute XPath
- B. Using accessibilityId or resource-id
- C. Using screen coordinates
- D. Using className only

Skill: Appium

Subskill: Locator reliability

Competency: Improving test stability

Difficulty Level: Advanced

Bloom Level: Evaluation

- **Correct Answer Reason:**
Option B is correct because accessibilityId and resource-id are stable and less prone to UI changes.
 - **Incorrect Options Feedback:**
 - o A: Absolute XPath is fragile.
 - o C: Coordinates are unreliable across devices.
 - o D: className may match multiple elements.
-

 **Topic 17: Appium Real Devices vs Emulators — Setup Differences, Challenges**

Multiple Choice Question 1

Which of the following is a common challenge when testing on real devices?

Options:

- A. Inconsistent UI rendering
- B. Lack of hardware sensors
- C. Device connection and authorization
- D. Unsupported gestures

Skill: Appium

Subskill: Device setup

Competency: Identifying real-device challenges

Difficulty Level: Intermediate

Bloom Level: Analysis

- **Correct Answer Reason:**
Option C is correct because real devices often require USB debugging and authorization for Appium to connect.
 - **Incorrect Options Feedback:**
 - o A: UI rendering is usually more accurate on real devices.
 - o B: Real devices have full sensor support.
 - o D: Gestures are supported if properly scripted.
-

Multiple Choice Question 2

Which tool is commonly used to connect Android real devices to Appium?

Options:

- A. Xcode
- B. ADB
- C. UIAutomator
- D. Espresso

Skill: Appium

Subskill: Android setup

Competency: Connecting devices

Difficulty Level: Intermediate

Bloom Level: Knowledge

- **Correct Answer Reason:**
Option B is correct because ADB (Android Debug Bridge) is used to connect and manage Android devices.
 - **Incorrect Options Feedback:**
 - o A: Xcode is for iOS development.
 - o C: UIAutomator is a test framework, not a connection tool.
 - o D: Espresso is used for Android testing, not device connection.
-

Multiple Choice Question 3

Which capability must be set to run tests on an emulator?

Options:

- A. `deviceName: "Android"`
- B. `platformVersion: "latest"`
- C. `avd: "<emulator_name>"`
- D. `realDevice: false`

Skill: Appium

Subskill: Emulator configuration

Competency: Setting desired capabilities

Difficulty Level: Advanced

Bloom Level: Application

- **Correct Answer Reason:**
Option C is correct because `avd` specifies the Android Virtual Device to launch.
- **Incorrect Options Feedback:**
 - o A: Too generic; doesn't specify emulator.
 - o B: `platformVersion` is required but not emulator-specific.
 - o D: `realDevice` is not a valid capability.

Topic 18: Appium — Capturing Screenshots During Tests

Multiple Choice Question 1

Which Appium method captures a screenshot of the current screen?

Options:

- A. `driver.getScreenshot()`
- B. `driver.captureScreen()`
- C. `driver.takeScreenshot()`
- D. `driver.getScreenshotAs(OutputType.FILE)`

Skill: Appium

Subskill: Screenshot capture

Competency: Capturing visual evidence

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option D is correct because `getScreenshotAs()` is the standard method used with WebDriver and Appium.
- **Incorrect Options Feedback:**
 - o A, B, C: These are not valid Appium methods.

Multiple Choice Question 2

When is it most useful to capture screenshots in Appium tests?

Options:

- A. Before launching the app
- B. After every test step
- C. On test failure or assertion error
- D. During capability setup

Skill: Appium

Subskill: Test diagnostics

Competency: Using screenshots effectively

Difficulty Level: Intermediate

Bloom Level: Evaluation

- **Correct Answer Reason:**
Option C is correct because screenshots help diagnose failures and provide visual logs.
- **Incorrect Options Feedback:**
 - o A: App is not yet rendered.
 - o B: May be excessive and slow down tests.
 - o D: Setup phase doesn't involve UI.

Multiple Choice Question 3

How can screenshots be integrated into test reports?

Options:

- A. By attaching them to ExtentReports or Allure
- B. By printing them to console
- C. By saving them in the APK
- D. By embedding them in desired capabilities

Skill: Appium

Subskill: Reporting

Competency: Enhancing test reports

Difficulty Level: Advanced

Bloom Level: Application

- **Correct Answer Reason:**
Option A is correct because ExtentReports and Allure support screenshot embedding for failed steps.
- **Incorrect Options Feedback:**
 - o B: Console cannot display images.

- o C: APK is static and not used for logging.
 - o D: Capabilities are for configuration, not reporting.
-

✓ Topic 19: Appium — Reporting Integration (ExtentReports/Allure with Appium)

Multiple Choice Question 1

Which ExtentReports class is used to log test steps in Appium?

Options:

- A. ExtentLogger
- B. ExtentTest
- C. ExtentReporter
- D. TestLogger

Skill: Appium

Subskill: Reporting

Competency: Logging test actions

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option B is correct because ExtentTest is used to log individual test steps and outcomes.
 - **Incorrect Options Feedback:**
 - o A: Not a valid ExtentReports class.
 - o C: ExtentReporter configures output, not logging.
 - o D: Not part of ExtentReports API.
-

Multiple Choice Question 2

Which lifecycle annotations are best for initializing and flushing ExtentReports in Appium with TestNG?

Options:

- A. @BeforeTest and @AfterTest
- B. @BeforeSuite and @AfterSuite
- C. @BeforeMethod and @AfterMethod
- D. @BeforeClass and @AfterClass

Skill: Appium
Subskill: Reporting integration
Competency: Managing report lifecycle
Difficulty Level: Advanced
Bloom Level: Application

- **Correct Answer Reason:**
Option B is correct because initializing and flushing reports at the suite level ensures complete coverage.
 - **Incorrect Options Feedback:**
 - o A: May not cover all tests in the suite.
 - o C: Used for per-test logging, not full lifecycle.
 - o D: Class-level setup may miss suite-wide context.
-

Multiple Choice Question 3

Which Allure annotation is used to attach screenshots in Appium?

Options:

- A. @Step
- B. @Attachment
- C. @Screenshot
- D. @Log

Skill: Appium
Subskill: Allure reporting
Competency: Enhancing reports with visuals
Difficulty Level: Advanced
Bloom Level: Application

- **Correct Answer Reason:**
Option B is correct because @Attachment allows embedding screenshots and other artifacts in Allure reports.
 - **Incorrect Options Feedback:**
 - o A: @Step defines logical steps, not attachments.
 - o C: Not a valid Allure annotation.
 - o D: @Log is not part of Allure.
-

 **Topic 20: Appium — Common Pitfalls & Troubleshooting (ADB, Driver Compatibility, Flakiness)**

Multiple Choice Question 1

Which of the following is a common cause of flaky Appium tests?

Options:

- A. Using accessibilityId for all elements
- B. Using hardcoded waits like Thread.sleep()
- C. Using explicit waits with ExpectedConditions
- D. Using stable locators like resource-id

Skill: Appium

Subskill: Test reliability

Competency: Identifying flakiness sources

Difficulty Level: Advanced

Bloom Level: Evaluation

- **Correct Answer Reason:**
Option B is correct because hardcoded waits are unreliable and lead to timing issues.
 - **Incorrect Options Feedback:**
 - o A: accessibilityId is stable.
 - o C: Explicit waits improve reliability.
 - o D: resource-id is a recommended locator.
-

Multiple Choice Question 2

Which tool helps verify if an Android device is properly connected for Appium testing?

Options:

- A. Android Studio
- B. Appium Inspector
- C. ADB
- D. UIAutomator

Skill: Appium

Subskill: Device connectivity

Competency: Troubleshooting device setup

Difficulty Level: Intermediate

Bloom Level: Application

- **Correct Answer Reason:**
Option C is correct because ADB (adb devices) lists connected devices and their status.
 - **Incorrect Options Feedback:**
 - o A: Useful for development, not direct connectivity check.
 - o B: Used for UI inspection.
 - o D: A test framework, not a connectivity tool.
-

Multiple Choice Question 3

Which capability mismatch often causes Appium session failures?

Options:

- A. Incorrect app package
- B. Missing screenshot path
- C. Wrong test name
- D. Unused environment variable

Skill: Appium

Subskill: Troubleshooting

Competency: Diagnosing session failures

Difficulty Level: Advanced

Bloom Level: Analysis

- **Correct Answer Reason:**
Option A is correct because an incorrect app package prevents Appium from launching the app.
 - **Incorrect Options Feedback:**
 - o B: Screenshot path doesn't affect session start.
 - o C: Test name is metadata.
 - o D: Environment variables are optional.
-