

TOP 100 CUCUMBER FRAME WORK INTERVIEW QUESTIONS AND ANSWERS

What is Cucumber?

- Cucumber is an open-source testing tool that supports Behavior Driven Development (BDD). It allows the execution of feature documentation written in plain language.
-

What are the main advantages of using Cucumber?

- Easy to understand for non-programmers
 - Bridges the gap between technical and non-technical team members
 - Supports multiple programming languages
 - Facilitates test automation
-

What are Gherkin keywords?

- Given, When, Then, And, But, Background, Scenario, Scenario Outline, Examples.
-

What is a Feature file?

- A Feature file contains the high-level description of a software feature and its scenarios, written using Gherkin language.
-

What is a Scenario in Cucumber?

- A Scenario represents a single behavior or example that needs to be tested.
-

What is a Step Definition in Cucumber?

- Step Definitions are the implementation of the steps in a scenario, written in a programming language like Java, Python, Ruby, etc.
-

What is a Scenario Outline?

- A Scenario Outline is used to execute the same Scenario multiple times with different sets of test data.
-

What is the purpose of the Examples table in a Scenario Outline?

- The Examples table provides the different data sets for executing the Scenario Outline.
-

What is Background in Cucumber?

- Background is used to define steps that are common to all scenarios in a feature file.
-

What are Tags in Cucumber?

- Tags are used to filter scenarios for execution. They can be applied to features, scenarios, or steps.
-

How do you set up Cucumber in a Java project?

- Add Cucumber dependencies in the build tool (Maven or Gradle), create feature files, and write step definitions.
-

What dependencies are needed to set up Cucumber with Maven?

- Cucumber-java, cucumber-Junit, and optionally cucumber-spring.
-

What is the purpose of the cucumber.options in a Cucumber project?

- cucumber.options is used to specify various options like the path to feature files, step definitions, tags, etc.
-

How do you run Cucumber tests with JUnit?

- Annotate a test runner class with `@RunWith(Cucumber.class)` and specify the `@CucumberOptions`.
-

What is a Cucumber Runner class?

- A Runner class is used to run the Cucumber tests. It configures the path to the feature files and step definitions.

How do you write a simple Scenario in Gherkin?

```
Feature: Login
  Scenario: Successful login with valid credentials
    Given User is on login page
    When User enters valid username and password
    Then User should be redirected to the dashboard
```

What is the purpose of Given, When, Then in a Scenario?

- Given sets up the initial context, when describes an action, and Then describes an expected outcome.
-

How do you handle complex steps in Gherkin?

- Break down the steps into smaller, more manageable steps or use Scenario Outline for repetitive steps.
-

Can you use AND and BUT in Gherkin?

- Yes, And and But can be used to add more steps to Given, When, or Then.
-

What is the purpose of the Background keyword?

- Background is used for steps that are common to all scenarios in a feature file.
-

How do you write a Step Definition in Java?

```
@Given("^User is on login page$")
public void user_is_on_login_page() {
    // Code to navigate to login page
}
```

How do you pass parameters to a Step Definition?

- Use placeholders in the step definition regex and method parameters.

```
@When("^User enters username (.+) and password (.+)$")
public void user_enters_credentials(String username, String password) {
    // Code to enter username and password
}
```

What is the use of the @Before and @After annotations in Cucumber?

- @Before and @After hooks are used to execute code before and after each scenario.
-

How do you handle exceptions in Step Definitions?

- Use try-catch blocks to handle exceptions and log errors.

Can you share state between Step Definitions? How?

- Yes, by using dependency injection frameworks like Spring or PicoContainer.
-

What are Hooks in Cucumber?

- Hooks are blocks of code that can be executed before or after each scenario, similar to setup and teardown methods.
-

How do you write a Before Hook in Cucumber?

```
@Before
public void setUp() {
    // Code to set up test data or environment
}
```

How do you write an After Hook in Cucumber?

```
@After
public void tearDown() {
    // Code to clean up test data or environment
}
```

Can Hooks be tagged in Cucumber? How?

- Yes, hooks can be tagged using @Before or @After with tag expressions.

```
@Before("@SmokeTest")
public void beforeSmokeTest() {
    // Code to execute before scenarios tagged with @SmokeTest
}
```

What is the use of @BeforeStep and @AfterStep hooks?

- @BeforeStep and @AfterStep hooks run before and after each step in a scenario.

How do you use Data Tables in Cucumber?

- Data Tables are used to pass a list of values or a table of values to a step definition.

```
When User enters the following details:
| username | password |
| user1    | pass1    |
| user2    | pass2    |
```

How do you handle Data Tables in Step Definitions?

- Use List<Map<String, String>> or DataTable parameter types.

```
@When("^User enters the following details:$")
public void user_enters_details(DataTable dataTable) {
    List<Map<String, String>> userDetails = dataTable.asMaps(String.class, String)
    for (Map<String, String> user : userDetails) {
        // Code to enter user details
    }
}
```

What is the difference between Data Tables and Examples tables?

- Data Tables are used for steps within a scenario, while Examples tables are used for Scenario Outlines to provide test data.

Can Data Tables handle complex data structures?

- Yes, Data Tables can represent complex data structures like nested lists and maps.
-

How do you convert a Data Table to a List of custom objects?

- Use the DataTable methods to convert the table directly.

```
List<User> users = dataTable.asList(User.class);
```

How do you integrate Cucumber with Selenium?

- Set up Selenium WebDriver in the step definitions to perform browser automation.

How do you handle browser setup and teardown in a Cucumber project?

- Use @Before and @After hooks to initialize and close the browser.
-

How do you capture screenshots in Cucumber?

- Use WebDriver's `getScreenshotAs` method in an `@After` hook.

Example:

`@After`

`public void`

`takeScreenshot(Scenario`

`scenario) if (scenario.isFailed()) {`

`final byte[] screenshot =`

`((TakesScreenshot)driver).getScreenshotAs(OutputType.BYTES);`

`scenario.attach(screenshot, "image/png", "screenshot");`

`}`

`}`

How do you handle synchronization issues in Selenium with Cucumber?

- Use explicit waits (`WebDriverWait`) to handle synchronization issues.
-

Can you use Page Object Model with Cucumber? How?

- Yes, create Page Object classes and use them in step definitions to separate test code from page-specific code.
-

What are different types of reports in Cucumber?

- HTML reports, JSON reports, JUnit reports, and custom reports using plugins.
-

How do you generate an HTML report in Cucumber?

- Use the `html:target/cucumber-html-report` plugin in `@CucumberOptions`.

```
@CucumberOptions(  
    plugin = {"html:target/cucumber-html-report"}  
)
```

What is the use of the JSON report in Cucumber?

- JSON report is used for further processing or integration with other tools like Jenkins or Allure.
-

How do you integrate Cucumber reports with Jenkins?

- Install the Cucumber Reports plugin in Jenkins.
 - Configure the Jenkins job to execute the Cucumber tests and set the path to the JSON report.
 - Add a post-build action to publish the Cucumber report.
-

How do you configure multiple report formats in Cucumber?

- Use the plugin option in `@CucumberOptions` to specify multiple report formats.

Example:

```
@CucumberOptions(plugin = {"html:target/cucumber-html-report",  
"json:target/cucumber.json", "junit:target/cucumber.xml"})
```

What is the advantage of using the Cucumber Extent Reports plugin?

- It provides rich HTML reports with detailed logs, screenshots, and various other features.
-

How do you add custom logs to Cucumber reports?

- Use the log method of the scenario object in step definitions.

Example: `scenario.log("Custom log message");`

How do you generate an Allure report for Cucumber tests?

- Add Allure Cucumber dependencies to the project.
- Configure the Allure report generation in the `@CucumberOptions`.
- Run the tests and generate the report using Allure command-line tools.

Why is it important to keep feature files concise and focused?

- It improves readability, maintainability, and ensures that the scenarios are easy to understand for all stakeholders.
-

How can you ensure that your Gherkin scenarios are maintainable?

- Keep the steps DRY (Don't Repeat Yourself), use Background wisely, and avoid hardcoding data.
-

How do you handle common test setup code in Cucumber?

- Use @Before hooks for common setup code and Background in feature files for common steps.
-

Why should you avoid using too many tags in Cucumber?

- It can lead to complex test execution configurations and make it harder to manage the scenarios.
-

What is the benefit of using Page Object Model (POM) with Cucumber?

- It promotes code reusability, readability, and separation of concerns by encapsulating page-specific logic.
-

How do you handle test data in Cucumber?

- Use external data sources like CSV, Excel, or databases, and parameterize the scenarios to make them data-driven.
-

What are custom parameter types in Cucumber?

- Custom parameter types allow you to define your own parameter types and transformations for steps.

Example:

```
ParameterType("currency", "USD|EUR|GBP", Currency.class, (String s) -> new  
Currency(s));
```

How do you reuse step definitions across different feature files?

- Place the step definitions in a common package that is accessible to all feature files.
-

What are the different ways to share state between step definitions?

- Use shared variables in a context class, dependency injection, or use static variables (though not recommended).
-

How do you handle asynchronous operations in Cucumber?

- Use synchronization mechanisms like explicit waits or callbacks to ensure the steps wait for operations to complete.
-

What is the use of the @BeforeAll and @AfterAll hooks?

- These hooks run once before all scenarios and once after all scenarios, respectively.
-

How do you implement dependency injection in Cucumber?

- Use frameworks like Spring, PicoContainer, or Guice to manage dependencies and share state.
-

How do you integrate Cucumber with JIRA?

- Use plugins or tools like Xray or Zephyr to link Cucumber scenarios with JIRA issues and track test results.
-

How do you integrate Cucumber with TestNG?

- Use the cucumber-testng library and configure the Cucumber runner with TestNG annotations.

Can you use Cucumber with REST Assured for API testing?

- Yes, combine Cucumber for BDD and REST Assured for API testing in the step definitions.

How do you manage configuration properties in a Cucumber project?

- Use a configuration file (like config.properties) and load it in step definitions or hooks.

What is the role of the @Transform annotation in Cucumber?

- @Transform is used to apply custom transformations to step parameters.
-

What do you do if a step definition is not found?

- Ensure the step definition regex matches the step in the feature file, and check the glue path in @CucumberOptions.
-

How do you debug Cucumber step definitions?

- Use breakpoints and a debugger in your IDE, or add logging statements to the step definitions.
-

What is the undefined step error, and how do you fix it?

- It occurs when a step definition is not found. Ensure the step text matches the regex in the step definition.
-

How do you handle flaky tests in Cucumber?

- Identify the cause (e.g., timing issues, environment issues) and use retries, better synchronization, or refactor tests.
-

How do you skip a Cucumber scenario?

- Use tags and configure the Cucumber runner to exclude the tagged scenarios.
-

What is BDD (Behavior Driven Development)?

- BDD is an agile software development process that encourages collaboration among developers, QA, and non-technical stakeholders to create shared understanding of requirements.
-

How does Cucumber support BDD?

- Cucumber allows writing tests in a natural language format that non-technical stakeholders can understand, fostering collaboration.
-

What are the limitations of Cucumber?

- Can be verbose for complex scenarios, requires collaboration to write meaningful feature files, and might not be suitable for all types of testing.
-

How do you handle large Cucumber projects?

- Organize feature files and step definitions into logical packages, use tags to manage test execution, and keep feature files concise.
-

What is the difference between TDD and BDD?

- TDD (Test Driven Development) focuses on writing tests before code, primarily for developers. BDD (Behavior Driven Development) focuses on the behavior of the application, involving collaboration between developers, testers, and business stakeholders.
-

How do you set up Cucumber tests to run in a CI/CD pipeline?

- Integrate Cucumber with a CI tool like Jenkins, GitLab CI, or CircleCI. Configure the pipeline to run Cucumber tests and generate reports.
-

How do you manage test environments for Cucumber tests in CI/CD?

- Use environment variables, configuration files, and CI/CD tool-specific environment management features.
-

What is parallel execution in Cucumber, and how do you achieve it?

- Parallel execution allows running multiple Cucumber scenarios or features simultaneously to reduce test execution time. Achieve it using plugins like cucumber-parallel, or frameworks like TestNG with Cucumber.
-

How do you handle test data setup and teardown in a CI/CD pipeline?

- Use @Before and @After hooks for test data setup and teardown. Manage test data using scripts or database tools in the pipeline.
-

How do you ensure Cucumber tests are idempotent in a CI/CD environment?

- Ensure tests do not depend on the state left by previous tests, use unique data for each run, and clean up test data after execution.
-

What are Cucumber Expressions?

- Cucumber Expressions are a simpler alternative to regular expressions for matching steps in Cucumber.
-

How do you write a Cucumber Expression?

- Use curly braces to denote parameters.

Example: When User enters {string} and {string}

What are optional text parameters in Cucumber Expressions?

- Optional text parameters allow matching text that may or may not be present in a step.

Example: When User enters {string} and {string} in the {string?} field

How do you use regular expressions in Cucumber step definitions?

- Use the @Given, @When, and @Then annotations with a regex pattern.

Example:

```
@When("^User enters (.+) and (.+)$")

public void user_enters(String username, String password) {

    // Code to enter credentials

}
```

What is the difference between Cucumber Expressions and regular expressions?

- Cucumber Expressions are simpler and more readable, designed for business-readable scenarios, while regular expressions are more powerful but complex.
-

What is the @Transform annotation used for in Cucumber?

- @Transform is used to apply custom transformations to step parameters.

Example: @Transform(DateConverter.class) Date date

How do you create a custom transformer in Cucumber?

- Implement the Transformer interface and override the transform method.

Example:

public class DateConverter extends

Transformer<Date> { @Override

public Date transform(String value) {

return new SimpleDateFormat("yyyy-MM-dd").parse(value);

}

How do you handle custom data transformations for complex types in Cucumber?

- Create a custom transformer class that implements the ParameterType interface.

```
public class ComplexTypeTransformer extends Transformer<ComplexType> {  
    @Override  
    public ComplexType transform(String value) {  
        // Custom transformation logic  
        return new ComplexType(value);  
    }  
}
```

What is the purpose of the @ParameterType annotation in Cucumber?

- @ParameterType is used to define custom parameter types that can be reused in step definitions.

```
@ParameterType("customType")  
public CustomType customType(String value) {  
    return new CustomType(value);  
}
```

How do you use a custom parameter type in a step definition?

- Define the custom parameter type and use it in the step definition.

Example:

```
@Given("^a custom type parameter (customType)$")  
  
public void a_custom_type_parameter(CustomType customType) {  
  
    // Use the customType parameter  
  
}
```

How do you ensure that Cucumber tests are robust in a CI/CD pipeline?

- Write reliable, independent, and idempotent tests.
 - Use mocks and stubs for unstable or external dependencies.
 - Implement proper setup and teardown procedures to avoid state leakage.
-

How can you execute Cucumber tests in parallel in a CI/CD pipeline?

- Use tools or plugins that support parallel execution, like Maven Surefire or TestNG with Cucumber.
 - Configure the CI/CD pipeline to distribute tests across multiple agents or nodes.
-

How do you handle environment-specific configurations in Cucumber tests?

- Use environment variables or configuration files to manage different environments.
 - Load environment-specific properties in the @Before hook.
-

How do you use Cucumber for mobile testing?

- Integrate Cucumber with Appium for mobile automation.
 - Write step definitions using Appium's API to interact with mobile devices.
-

How do you set up Cucumber with Appium for mobile automation?

- Add Cucumber and Appium dependencies to the project.
 - Configure Appium server and desired capabilities.
 - Implement step definitions using Appium to perform actions on the mobile app.
-

How do you manage different mobile devices and platforms in Cucumber?

- Use tags to differentiate tests for various devices and platforms.
 - Configure Appium capabilities dynamically based on the device/platform.
-

Can you integrate Cucumber with JUnit 5? How?

- Yes, use the @Cucumber annotation provided by cucumber-junit-platform-engine.
 - Configure the JUnit 5 platform in the @CucumberOptions.
-

How do you integrate Cucumber with Serenity BDD?

- Add Serenity BDD dependencies to the project.
 - Use the @CucumberOptions annotation to specify Serenity plugins.
 - Write step definitions using Serenity BDD features for better reporting and test management.
-

What is the benefit of integrating Cucumber with RestAssured for API testing?

- Allows for writing BDD-style tests for REST APIs.
 - Simplifies testing by combining Cucumber's readability with RestAssured's powerful API testing capabilities.
-

How do you implement BDD for a microservices architecture using Cucumber?

- Write feature files that describe the interactions between microservices.
 - Implement step definitions to test each microservice independently.
 - Use mocks and stubs to simulate interactions with other services.
-



Final Tips for Cucumber

What are some common pitfalls to avoid when using Cucumber?

- Writing overly detailed steps that obscure the behavior being described.
 - Failing to collaborate with non-technical stakeholders when writing feature files.
 - Not maintaining feature files and step definitions, leading to outdated tests.
-

How can you improve the performance of Cucumber tests?

- Optimize the setup and teardown processes.
- Use parallel execution where possible.
- Refactor step definitions to reduce redundancy and improve efficiency.

What strategies can be used to maintain large Cucumber test suites?

- Regularly review and refactor feature files and step definitions.
 - Group related scenarios and features logically.
 - Use tags to organize and selectively execute tests.
-

How do you ensure Cucumber tests are up-to-date with the latest application changes?

- Foster continuous collaboration between developers, testers, and business stakeholders.
- Regularly update feature files to reflect changes in application behavior.
- Integrate Cucumber tests into the CI/CD pipeline for immediate feedback on changes.

