BDD Cucumber JAVA

Press Space for next page →

Agenda

1. Introduction to BDD
2. Key Concepts of BDD
3. Introduction to Cucumber
4. Gherkin Syntax
5. Example of Gherkin Syntax
6. Benefits of BDD with Cucumber
7. Setting Up Cucumber
8. Add Cucumber Dependencies
9. Create Feature Files
10. Implement Step Definitions
11. Run Cucumber Tests

Introduction to BDD

Definition:

BDD is a software development approach that emphasizes collaboration between developers, QA, and non-technical or business participants in a software project.

Purpose:

To improve communication, create shared understanding, and ensure the software meets business requirements.

Key Concepts of BDD

User Stories:

Descriptions of features from an end-user perspective.

Scenarios:

Detailed examples of user stories, outlining specific use cases.

Gherkin Syntax:

A business-readable domain-specific language for describing software behavior.

Introduction to Cucumber

Definition:

Cucumber is a tool for running automated tests written in plain language.

Language:

Uses Gherkin syntax to define test cases.

Integration:

Can be integrated with various programming languages like Java, Ruby, and JavaScript.

Gherkin Syntax

Feature: Describes the feature under test.
Scenario: Describes a specific use case.
Steps: Given, When, Then, And, But.

- Background: Common steps for all scenarios.
- Scenario Outline: Parameterized scenarios.
- Examples: Data tables for Scenario Outline.
- Given: Sets up the initial state.
- When: TDescribes an action.
- Then: Describes an expected outcome.
- And: Additional steps.

Example of Gherkin Syntax

```
Feature: Login Functionality
    Scenario: Valid Login
        Given User is on the login page
        When User enters valid username
        And User enters valid password
        Then User should be logged in successfully
```

```
Feature: User Login
    Scenario Outline: Login with multiple credentials
        Given User is on the login page
        When User enters "<username>" and "<password>"
        Then User should see "<result>"

    Examples:
        | username   | password | result                 |
        | user1      | pass123  | Welcome message        |
        | user2      | wrongpwd | Error: Invalid password |
        | invalidusr | pass123  | Error: Invalid username |
```

Benefits of BDD with Cucumber

Improved Communication: Common language understood by all stakeholders.
Clear Requirements: Detailed and executable specifications.
Early Bug Detection: Automated tests catch issues early.
Living Documentation: Tests serve as up-to-date documentation.

Setting Up Cucumber

- **Install Cucumber:** Follow installation steps for your programming language.
- **Write Feature Files:** Use Gherkin syntax to create feature files.
- **Implement Step Definitions:** Code to automate the steps defined in feature files.
- **Run Tests:** Use Cucumber to execute the tests and validate behavior.

Add Cucumber Dependencies

Link to the Cucumber Java and Cucumber TestNG dependencies.

Add the following dependencies to your `pom.xml` file for Maven projects:

```
<dependency>
    <groupId>io.cucumber</groupId>
    <artifactId>cucumber-java</artifactId>
    <version>7.20.1</version>
</dependency>
<dependency>
    <groupId>io.cucumber</groupId>
    <artifactId>cucumber-testng</artifactId>
    <version>7.20.1</version>
</dependency>
```

Create Feature Files

Create feature files with `.feature` extension using Gherkin syntax in the `src/test/resources/features` directory.

```
Feature: Login to SWAG Labs
    Scenario: Login with valid credentials
        Given User is on the login page
        When User enters username "standard_user" and password as "secret_sauce"
        Then User should be logged in successfully
```

Feature files for multiple data sets can use Scenario Outline and Examples.

```
Feature: Login to SWAG Labs
    Scenario Outline: Login with multiple credentials
        Given User is on the login page
        When User enters "<username>" and "<password>"
        Then User should see "<result>"

    Examples:
        | username              | password      | result       |
        | standard_user         | secret_sauce  | Products     |
        | locked_out_user       | secret_sauce  | Products     |
```

Implement Step Definitions

Create step definition classes to map Gherkin steps to Java code. These classes should be in the `src/test/java/stepdefinitions` package.

```
class LoginPageDefinitions {
    @Given("User is on the login page")
    public void userIsOnLoginPage() {
        // Code to navigate to the login page
    }

    @When("User enters username {string} and password as {string}")
    public void userEntersCredentials(String username, String password) {
        // Code to enter username and password
    }

    @Then("User should be logged in successfully")
    public void userShouldBeLoggedIn() {
        // Code to verify successful login
    }
}
```

Run Cucumber Tests

Create a test runner class to execute the Cucumber tests. This class should be in the `src/test/java/runners` package.

```
@CucumberOptions(features = "src/test/resources/features",
                 glue = "stepdefinitions",
                 plugin = {})
public class CucumberRunnerTests extends AbstractTestNGCucumberTests {

}
```

Thank you ❤️

qa-june-2024-automation-with-java-slides