BDD

Lesson 03 : Gherkin Language – Features, Scenario, Scenario Outline

Capgemini

Lesson Objectives

To understand the following topics:

- Feature
- Background
- Scenario
  - Given Keyword
  - When Keyword
  - Then Keyword
  - And Keyword
  - But Keyword
  - * Keyword
- Scenario Outline

Feature

- Each *Gherkin* file begins with a ***Feature*** keyword.
- *Feature* defines the logical test functionality you will test in this feature file.
- For e.g
  - If you are testing a payment gateway your *Feature*will become *Payment Gateway* or
  - if you are testing the *LogIn*functionality then the *Feature* will become *Login*.

- The idea of having a feature file is to put down a summary of what you will be testing. This will serve as the documentation for your tests as well as a good point to start for a new team member.
- Note that a feature keyword is present at the starting of the feature file.

Feature

***Feature**: LogIn Action Test*

Or

***Feature**: LogIn Action Test*
Description: This feature will test a LogIn and LogOut functionality

Or

***Feature**: LogIn Action Test*
This feature will test a LogIn and LogOut functionality

- Notice that whatever comes after the ***Feature: keyword,*** will be considered as the feature description.
- Feature description can span across multiple lines like shown above in second example.
- Everything after *Feature:* till the next Keyword is encountered is considered as feature description.

Background

- ***Background*** keyword is used to define steps which are common to all the tests in the feature file.

- For example to purchase a product, you need to do following steps:
  - *Navigate to Home Page*
  - *Click on the LogIn link*
  - *Enter UserName and Password*
  - *Click on Submit button*

  After these steps only you will be able to add a product to your *cart/basket* and able to perform the payment.

  Now as we are in a feature file where we will be testing only the *Add to Cart* functionality, t hese tests become common for all tests.

  So instead of writing them again and again for all tests we can move it under the background keyword.

Background

Example of background in Feature file

***Feature**: Add to Cart*

This feature will test functionality of adding different products to the User basket from different flow

***Background:** User is Logged In*

***Scenario**: Search a product and add the first result/product to the User basket*

***Given** User searched for Lenovo Laptop*

***When** Add the first laptop that appears in the search result to the basket*

***Then** User basket should display with 1 item*

Scenario

- Each Feature will contain some number of tests to test the feature.
- Each test is called a ***Scenario*** and is described using the *Scenario:* keyword.

> ***Scenario****: Search a product and add the first result/product to the User basket*
> *Or*
> ***Scenario****: Successful LogIn with Valid Credentials*

A scenario is equivalent to a test in our regular development process.

Each scenario/test can be basically broken down into three parts:

- ***Precondition*** *to the test, which represent with (****Given****) keyword*
- ***Test step*** *execution, which represent with (****When****) keyword*
- ***Verification*** *of the output with expected result, which represent with (****Then****)*

Given Keyword

Given defines a precondition to the test.

For e.g.

In shopping website, assume that the *LogIn page link* is only present on the Home Page, so the precondition for clicking the *LogIn link* is that the user is at the Home Page.

If user is not at the Home Page, user would not be able to enter *Username* & *Password*.

This precondition can be expressed in *Gherkin* like this:

***Scenario**: Successful LogIn with Valid Credentials*
***Given** User is on Home Page*
***When** User Navigate to LogIn Page*

When Keyword

- ***When*** keyword defines the test action that will be executed.
- By test action we mean the user input action

***Scenario**: Successful LogIn with Valid Credentials*
***Given** User is on Home Page*
***When** User Navigate to LogIn Page*

Here user is performing some action using *When* keyword, clicking on the LogIn link.

We can see that when defines the action taken by the user.

It’s the event that will cause the actual change in state of the application.

Then Keyword

- ***Then*** keyword defines the Outcome of previous steps.
- We can understand it best by looking at the test above and adding a Then step there.

***Feature**: LogIn Action Test*
Description: This feature will test a LogIn and LogOut functionality
***Scenario**: Successful Login with Valid Credentials*
    ***Given** User is on Home Page*
    ***When** User Navigate to LogIn Page*
    ***And** User enters UserName and Password*
    ***Then** Message displayed LogIn Successfully*

Here we can see that ***Then*** is the outcome of the steps above. T

he reader of this test would easily be able to relate to *Then* step and would understand that when the above conditions are fulfilled then the *Then* step will be executed.

And Keyword

- ***And*** keyword is used to add conditions to your steps.
- Let's look at it by modifying our example a little

***Feature**: LogIn Action Test*
Description: This feature will test a LogIn and LogOut functionality
***Scenario**: Successful Login with Valid Credentials*
    ***Given** User is on Home Page*
    ***When** User Navigate to LogIn Page*
    ***And** User enters UserName and Password*
    ***Then** Message displayed Login Successfully*

Here we can see that ***Then*** is the outcome of the steps above.

The reader of this test would easily be able to relate to *Then* step and would understand that when the above conditions are fulfilled then the *Then* step will be executed.

And Keyword

***Feature**: LogIn Action Test*
Description: This feature will test a LogIn and LogOut functionality
***Scenario**: Successful Login with Valid Credentials*
    ***Given** User is on Home Page*
    ***And** LogIn Link displayed*
    ***When** User Navigate to LogIn Page*
    ***And** User enters UserName and Password*
    ***Then** Message displayed Login Successfully*
    ***And** LogOut Link displayed*

- Here you would see that *And* is being used to add more details to the *Given* step,
- it's simply adding more conditions. We have just added three conditions.
- Use it when you have specified more than one condition.
- *And* is used to add more conditions to *Given*, *When* and *Then* statements.

But Keyword

- ***But*** keyword is used to add negative type comments.
- It is not a hard & fast rule to use but only for negative conditions.
- It makes sense to use *But* when you will try to add a condition which is opposite to the premise your test is trying to set.
- Take a look at the example below:

***Feature**: LogIn Action Test*
Description: This feature will test a LogIn and LogOut functionality
***Scenario**: Unsuccessful Login with InValid Credentials*
    ***Given** User is on Home Page*
    ***When** User Navigate to LogIn Page*
    ***And** User enters UserName and Password*
    ***But** The user credentials are wrong*
    ***Then** Message displayed Wrong UserName & Password*

Here you can see how adding ***But*** has helped define a negative test,

in this test we will try to test failure conditions.

Where a wrong credentials are a failure condition.

* Keyword

- This keyword defies the whole purpose of having Given, When, Then and all the other keywords.
- Basically Cucumber doesn’t care about what Keyword you use to define test steps, all it cares about what code it needs to execute for each step.
- At this time just remember that all the keywords can be replaced by the ***** ***keyword*** and your test will just work fine. Let’s see with example, we had this test earlier:

***Feature**: LogIn Action Test*
Description: This feature will test a LogIn and LogOut functionality
***Scenario**: Successful Login with Valid Credentials*
    ***** *User is on Home Page*
    ***** *User Navigate to LogIn Page*
    ***** *User enters UserName and Password*
    ***** *Message displayed Login Successfully*

Scenario Outline

- **Scenario outline** basically replaces variable/keywords with the value from the table. Each row in the table is considered to be a scenario.
- It is repetitive and cumbersome to write same scenarios again and again with different values.

Scenario: *Find if subtraction is working Given there are 10 apples*
When *I give 3 bananas to my friend*
Then *I should have 7 bananas*
Scenario: *Find if subtraction is working*
Given *there are 20 apples*
When *I give 5 bananas to my friend*
Then *I should have 15 bananas*

- You can re-write above two scenarios into one scenario by using ‘Scenario Outline’.
- You need to use replace Scenario with ‘Scenario Outline’.
- Convert all data values into a tabular format and use the column name as a reference using < > sign.

Scenario Outline

- Example

```
Scenario Outline: Find if subtraction is working Given there are <total> apples
When I give <donated> bananas to my friend
Then I should have <remaining> bananas
Example:
|total |donated  |remaining |
|10    |3        |7         |
|20    |5        |15        |
```

- This is very helpful if you are testing a field or form with various combinations of values.
- This is possible by placeholder and you can identify placeholder by text surrounded by <>.
- Placeholder work as a variable in programming.
- Code execute a number of times based on available rows in ‘Example’ keyword.
- Each time placeholder replaced by values given in corresponding column Name

Summary

- In this lesson we have learned
- Feature
- Background
- Scenario
  - Given Keyword
  - When Keyword
  - Then Keyword
  - And Keyword
  - But Keyword
  - * Keyword
- Scenario Outline

Review Question

Question 1

- Which of the below is not a Gherkin Keyword?
  A. But
  B. When
  C. Given
  D. Outline

Question 2: Scenario outline is created to use same scenario for multiple values.

A. True
B. False