This is very often required in any automated test to pass data or to use the same test again with different data set. And the good part is that the ***Cucumber*** inherently supports ***Data Driven Testing using Scenario Outline***. There are different ways to use the data insertion within the *Cucumber* and outside the *Cucumber* with external files.

Data-Driven Testing in Cucumber

Parameterization without Example Keyword

Data-Driven Testing in Cucumber using Scenario Outline

Parameterization with Example Keyword
Parameterization using Tables

Data-Driven Testing in Cucumber using External Files

Parameterization using Excel Files
Parameterization using Json
Parameterization using XML

Scenario Outline - This is used to run the same scenario for 2 or more different sets of test data. ***E.g***. In our scenario, if you want to register another user you can data drive the same scenario twice.

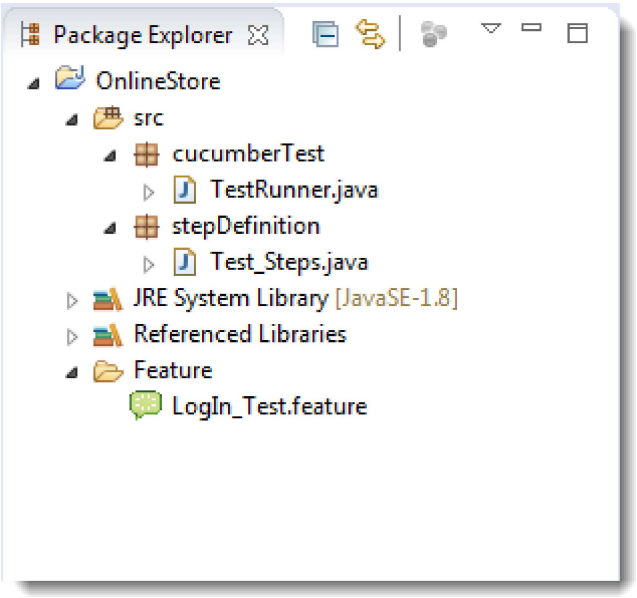
Examples - All scenario outlines have to be followed with the Examples section. This contains the data that has to be passed on to the scenario.

Data-Driven Testing in Cucumber

In the series of previous chapters, we are following the LogIn scenario. To demonstrate how parametrizing works, I am taking the same scenario again. It is important for you to be on the same page in term of project code, else you may get confused. Let's take a look at the current state of the project. In case you find it confusing, I would request you to go through the previous tutorial of ***Feature File***, ***Test Runner*** &***Step Definition***.

The project folder structure and code should be in the below state.

Package Explorer



LogIn_Test.fetaure

```
Feature: Login Action

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

Scenario: Successful LogOut
        When User LogOut from the Application
        Then Message displayed LogOut Successfully
```

Test_Steps.java

```
package stepDefinition;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

import cucumber.api.java.en.Given;
import cucumber.api.java.en.Then;
import cucumber.api.java.en.When;

public class Test Steps {
                public static WebDriver driver;
        @Given("^User is on Home Page$")
        public void user is on Home Page() throws Throwable {
                driver = new FirefoxDriver();
            driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
            driver.get("https://www.store.demoqa.com");
        }
```

```
	@When("^User Navigate to LogIn Page$")
	public void user Navigate to LogIn Page() throws Throwable {
		driver.findElement(By.xpath(".//*[@id='account']/a")).click();
	}

	@When("^User enters UserName and Password$")
	public void user enters UserName and Password() throws Throwable {
		driver.findElement(By.id("log")).sendKeys("testuser_1");
	    driver.findElement(By.id("pwd")).sendKeys("Test@123");
	    driver.findElement(By.id("login")).click();
	}

	@Then("^Message displayed Login Successfully$")
	public void message displayed Login Successfully() throws Throwable {
		System.out.println("Login Successfully");
	}

	@When("^User LogOut from the Application$")
	public void user LogOut from the Application() throws Throwable {
		driver.findElement (By.xpath(".//*[@id='account_logout']/a")).click();
	}

	@Then("^Message displayed LogOut Successfully$")
	public void message displayed LogOut Successfully() throws Throwable {
		System.out.println("LogOut Successfully");
	}
}
```

TestRunner.java

```
package cucumberTest;

import org.junit.runner.RunWith;
import cucumber.api.CucumberOptions;
import cucumber.api.junit.Cucumber;

@RunWith(Cucumber.class)
@CucumberOptions(
		features = "Feature"
		,glue={"stepDefinition"}
		)

public class TestRunner {

}
```

Parameterizing without Example Keyword

Now the task is to ***Parameterizing the UserName and Password***. Which is quite logical, why would anybody want to hardcode the *UserName* & *Password* of the application. As there is a high probability of changing both.

. Go to the ***Feature File*** and change the statement where passing *Username & Password* as per below:

And User enters "testuser_1" and "Test@123"

In the above statement, we have passed *Username & Password* from the *Feature File* which will feed in to *Step Definition* of the above statement automatically. *Cucumber* will do the trick for us. After the above changes, the code will look like this:

LogIn_Test.feature

```
Feature: Login Action

Scenario: Successful Login with Valid Credentials
        Given User is on Home Page
        When User Navigate to LogIn Page
        And User enters "testuser 1" and "Test@123"
        Then Message displayed Login Successfully

Scenario: Successful LogOut
        When User LogOut from the Application
        Then Message displayed LogOut Successfully
```

. Changes in the *Step Definition* file is also required to make it understand the *Parameterization of the feature file*. So, it is required to update the *Test Step* in the *Step Definition* file which is linked with the above-changed *Feature* file statement. Use the below code:

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

The same can be achieved by using the below code as well:

@When("^User enters "([^"])" and "([^"])"$")

With the help of the above statements, Cucumber will understand that the associated *Test_Step* is expecting some parameters.

. Same parameters should also go into the associated *Test_Step*. As the Test step is nothing but a simple Java method, syntax to accept the parameter in the Java method is like this:

public void user_enters_UserName_and_Password(String username, String password) throws Throwable {

}

. Now the last step is to feed the parameters in the actual core statements of *Selenium WebDriver*. Use the below code:

driver.findElement(By.id("log")).sendKeys(username);

driver.findElement(By.id("pwd")).sendKeys(password);

driver.findElement(By.id("login")).click();

After making the above changes, the method will look like this:

```
@When("^User enters \"(.*)\" and \"(.*)\"$")
        public void user enters UserName and Password(String username, String password) t
                driver.findElement(By.id("log")).sendKeys(username);
            driver.findElement(By.id("pwd")).sendKeys(password);
            driver.findElement(By.id("login")).click();
        }
```

. Run the test by *Right Click* on ***TestRunner class*** and Click ***Run As > JUnit Test*** Application. You would notice that the *Cucumber* will open the Website in the browser and enter *username & password* which is passed from the *Feature File*.

The next chapter is about doing ***Parameterization using Example Keyword in Cucumber***.