***Data Tables in Cucumber*** are quite interesting and can be used in many ways. *DataTables* are also used to handle large amounts of data. They are quite powerful but not the most intuitive as you either need to deal with a ***list of maps*** or a ***map of lists***. Most of the people get confused with Data tables & Scenario outline, but these two works completely differently.

## Difference between Scenario Outline & Data Table

***Scenario Outline:***

*This uses Example keyword to define the test data for the Scenario*
*This works for the whole test*
*Cucumber automatically run the complete test the number of times equal to the number of data in the Test Set*

***Test Data:***

*No keyword is used to define the test data*
*This works only for the single step, below which it is defined*
*A separate code needs to understand the test data and then it can be run single or multiple times but again just for the single step, not for the complete test*

As I said above, the *Data Tables* can be used in many ways because it has provided many different methods to use. Let's just go through a few most popular methods. I will choose a simple scenario to illustrate the working of the *Data Table* but we will make effective use of this when we will do ***Cucumber Framework*** in the next series of this ***Cucumber Tutorial***.

## Data Tables in Cucumber

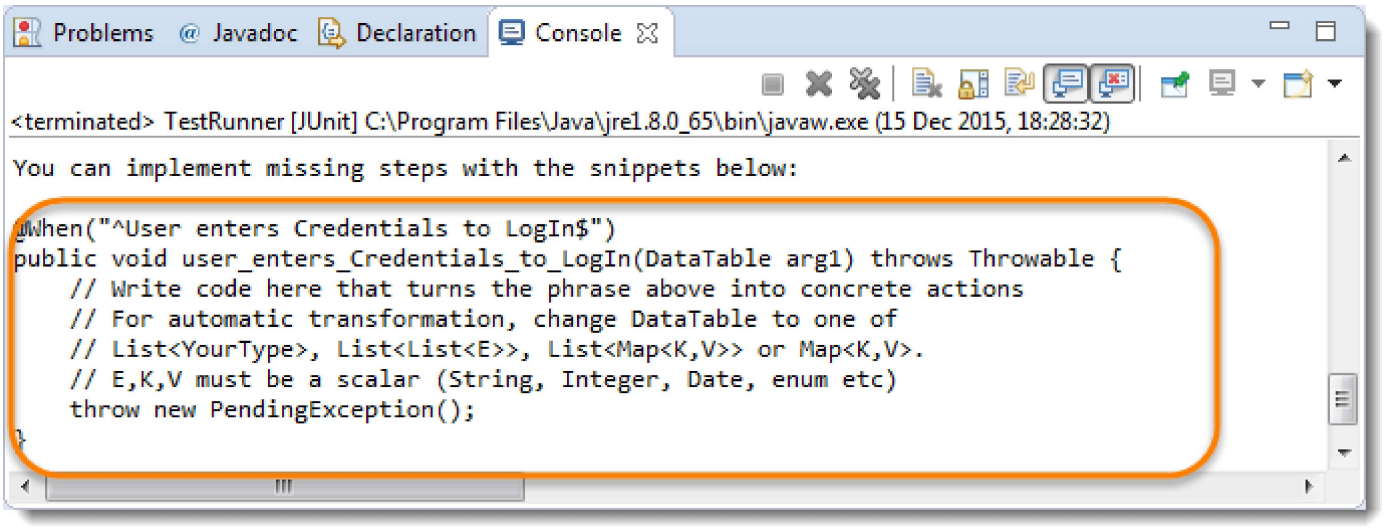
In this example, we will pass the test data using the *data table* and handle it using ***Raw()*** method.

```
Scenario: Successful Login with Valid Credentials
        Given User is on Home Page
        When User Navigate to LogIn Page
        And User enters Credentials to LogIn
    | testuser 1 | Test@153 |
        Then Message displayed Login Successfully
```

The complete scenario is same as what we have done earlier. But the only difference is in this, we are not passing parameters in the step line and even we are not using Examples test data. We

declared the data under the step only. So we are using Tables as arguments to Steps.

If you run the above scenario without implementing the step, you would get the following error in the Eclipse console window.




Copy the above hint in the *Step Definition* file and complete the implementation.

***The implementation of the above step will be like this:***

```
The implementation of the above step will belike this:
        @When("^User enters Credentials to LogIn$")
        public void user_enters_testuser__and_Test(DataTable usercredentials) throws Thro

                //Write the code to handle Data Table
                List<List<String>> data = usercredentials.raw();

                //This is to get the first data of the set (First Row + First Column)
                driver.findElement(By.id("log")).sendKeys(data.get(0).get(0));

                //This is to get the first data of the set (First Row + Second Column)
            driver.findElement(By.id("pwd")).sendKeys(data.get(0).get(1));

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

## *The complete test Implementation*

***Test Runner Class***

```
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 {

}
```

### *Feature File*

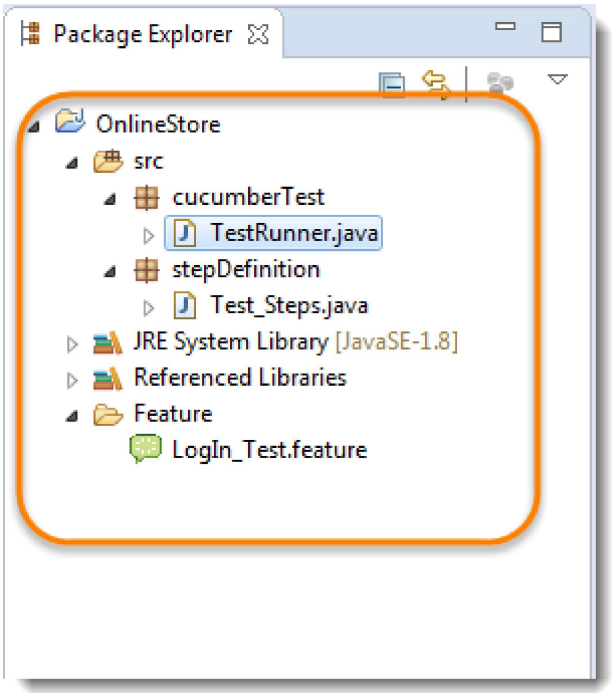
```
Feature: Login Action

Scenario: Successful Login with Valid Credentials
        Given User is on Home Page
        When User Navigate to LogIn Page
        And User enters Credentials to LogIn
    | testuser 1 | Test@153 |
        Then Message displayed Login Successfully

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

### *Step Definition*

```
package stepDefinition;

import java.util.List;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import cucumber.api.DataTable;
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 Credentials to LogIn$")
        public void user enters testuser  and Test(DataTable usercredentials) throws Thro
                List<List<String>> data = usercredentials.raw();
                driver.findElement(By.id("log")).sendKeys(data.get(0).get(0));
            driver.findElement(By.id("pwd")).sendKeys(data.get(0).get(1));
            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");

        }
}
```

***Project Explorer***




Run the test by *Right Click* on ***TestRunner class*** and Click ***Run As  > JUnit*** Test Application. you will notice that Cucumber will automatically figure out, what to provide in the Username and Password field.

In the next chapter of ***Data Tables in Cucumber using Maps***, we will handle little complex data.