Way2Automation - Tutorial 10 – Data Driven Testing in Cucumber

What you will Learn :

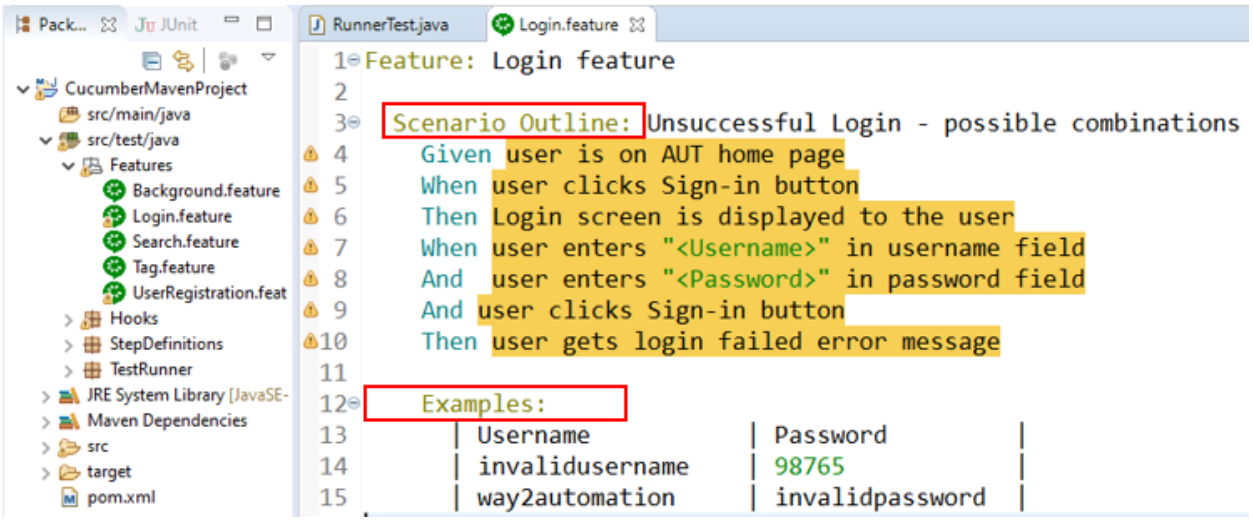
- Data Driven Testing using 'Examples' keyword plus 'Scenario Outline'

Data Driven Testing using 'Examples' keyword plus 'Scenario Outline'

Let us create a brand new feature file having 'Login feature'.

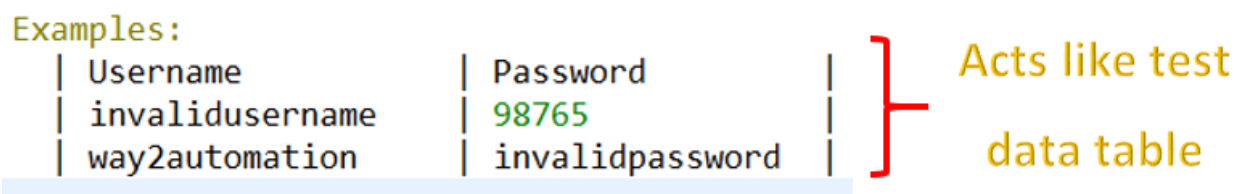
If you are writing 'Scenario Outline', you have to use 'Examples' keyword, see below. These 2 go hand-in-hand. You cannot use 'Examples' keyword with normal 'Scenario' that we have been writing so far.

So, if you really want to use the concept of data driven testing, you can achieve with the help of 'Examples' keyword and you have to use 'Scenario Outline'.



Save the file.

The 'Examples' table acts like a test data table for us. It contains multiple set of test data to test our application



Now, in the above 'Scenario Outline', look at line numbers 7 and 8. We have written "<Username>" and "<Password>". So these are string values because these are written within double quotes

```
7     When user enters "<Username>" in username field
8     And  user enters "<Password>" in password field
```

Now, the <Username> that you see in the figure above is actually the column name that we have written in the 'Examples' table. Similar is the case with <Password>. These should be an exact match. You cannot write <Username> in line number 7 and USERNAME in 'Examples' table.

```
Examples:
  | Username
  | invalidusername
  | way2automation
```

So the entire scenario will be executed 2 times since wehave 2 rows in our 'Examples' table

```
RunnerTest.java    Login.feature
1  Feature: Login feature
2
3    Scenario Outline: Unsuccessful Login - possible combinations
4      Given user is on AUT home page
5      When user clicks Sign-in button
6      Then Login screen is displayed to the user
7      When user enters "<Username>" in username field
8      And  user enters "<Password>" in password field
9      And user clicks Sign-in button
10     Then user gets login failed error message
11
12     Examples:
13       | Username          | Password         |
14       | invalidusername   | 98765            |
15       | way2automation    | invalidpassword  |
```

2 rows

So, this is the concept of data driven testing in cucumber.

Let us run our feature file to generate our step definition methods

```
Login.feature
1  Feature: Login feature
2
3    Scenario Outline: Unsuccessful Login - po
4      Given user is on AUT home page
5      When user clicks Sign-in button
6      Then Login screen is displayed to the u
7      When user enters "<Username>" in usern
8      And  user enters "<Password>" in passwo
9      And user clicks Sign-in button
10     Then user gets login failed error messa
11
12     Examples:
13       | Username          | Password
14       | invalidusername   | 98765
15       | way2automation    | invalidpasswor
16
```

```
Undo                    Ctrl+Z
Revert File
Save                    Ctrl+S
Open With               >
Show In                 Alt+Shift+W >
Cut                     Ctrl+X
Copy                    Ctrl+C
Paste                   Ctrl+V
Quick Fix               Ctrl+1
Shift Right
Shift Left
Pretty Format           Ctrl+Shift+F
Find Step               F3
Recalculate steps       F5
Coverage As             >
Run As                  >   1 Cucumber Feature
```

Copy the methods from console

```
Console
<terminated> Login.feature [Cucumber Feature] C:\Program Files\Java\jdk1.8.0_191\bin\javaw.exe (09-Mar-2021, 2:22:58 PM)
@Given("user is on AUT home page")
public void user_is_on_aut_home_page() {
    // Write code here that turns                                   crete actions
    throw new io.cucumber.java.Pe
}

@When("user clicks Sign-in button
public void user_clicks_sign_in_b
    // Write code here that turns                                   crete actions
    throw new io.cucumber.java.Pe
}

@Then("Login screen is displayed
public void login_screen_is_displ
    // Write code here that turns                                   crete actions
    throw new io.cucumber.java.Pe
}

@When("user enters {string} in username field")
public void user_enters_in_username_field(String string) {
    // Write code here that turns the phrase above into concrete actions
    throw new io.cucumber.java.PendingException();
}

@When("user enters {string} in password field")
public void user_enters_in_password_field(String string) {
    // Write code here that turns the phrase above into concrete actions
    throw new io.cucumber.java.PendingException();
```

```
Cut                     Ctrl+X
Copy                    Ctrl+C
Paste                   Ctrl+V
Select All              Ctrl+A
Find/Replace...
Open Link
Clear
Remove All Terminated
Terminate/Disconnect All
Scroll Lock
Word Wrap
Preferences...
```

Create a step definition file and paste the above methods

Login.feature Login.java

```java
package StepDefinitions;

public class Login {
    @Given("user is on AUT home page")
    public void user_is_on_aut_home_page() {
        // Write code here that turns the phrase above into concrete a
        throw new io.cucumber.java.PendingException();
    }

    @When("user clicks Sign-in button")
    public void user_clicks_sign_in_button() {
        // Write code here that turns the phrase above into concrete a
        throw new io.cucumber.java.PendingException();
    }

    @Then("Login screen is displayed to the user")
    public void login_screen_is_displayed_to_the_user() {
        // Write code here that turns the phrase above into concrete a
        throw new io.cucumber.java.PendingException();
    }

    @When("user enters {string} in username field")
    public void user_enters_in_username_field(String string) {
```

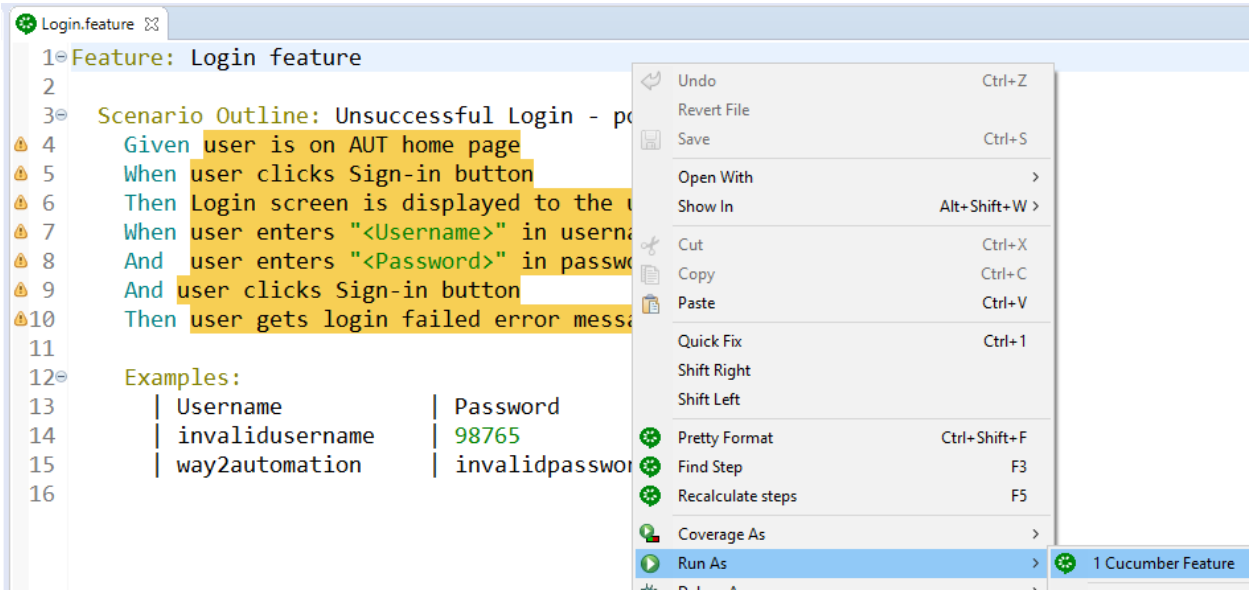
Import the methods and remove the exceptions

Login.feature Login.java

```java
package StepDefinitions;

import io.cucumber.java.en.Given;

public class Login {
    @Given("user is on AUT home page")
    public void user_is_on_aut_home_page() {

    }

    @When("user clicks Sign-in button")
    public void user_clicks_sign_in_button() {

    }

    @Then("Login screen is displayed to the user")
    public void login_screen_is_displayed_to_the_user() {

    }

    @When("user enters {string} in username field")
    public void user_enters_in_username_field(String string) {

    }

    @When("user enters {string} in password field")
```

Save the file

Let us re-run our feature file



Notice the console output. The scenario got executed twice. The username/password was picked up from the 'Examples' table.

Also, 14 steps got passed since we have 7 steps in a scenario. This would be multiplied by 2 rows in the table, hence 14 steps

```
Console
<terminated> Login.feature [Cucumber Feature] C:\Program Files\Java\jdk1.8.0_191\bin\javaw.exe (09-Mar-2021, 2

Scenario Outline: Unsuccessful Login - possible combinations
  Given user is on AUT home page
  When user clicks Sign-in button
  Then Login screen is displayed to the user
  When user enters "invalidusername" in username field
  And user enters "98765" in password field
  And user clicks Sign-in button
  Then user gets login failed error message

Scenario Outline: Unsuccessful Login - possible combinations
  Given user is on AUT home page
  When user clicks Sign-in button
  Then Login screen is displayed to the user
  When user enters "way2automation" in username field
  And user enters "invalidpassword" in password field
  And user clicks Sign-in button
  Then user gets login failed error message

2 Scenarios (2 passed)
14 Steps (14 passed)
```

So, invalidusername/98765 was picked up in first scenario execution and way2automation/invalidpassword was picked up in second scenario execution

```
Examples:
  | Username          | Password        |
  | invalidusername   | 98765           |
  | way2automation    | invalidpassword |
```

Let us now take another example wherein both the rows have only numeric values, see below

```
Invoice.feature
1  Feature: Calculate invoice amount
2
3    Scenario Outline: Invoice amount - possible combinations
4      Given user is on invoice page
5      When user enters invoice amount "<InvoiceAmount>"
6      When user enters tax amount "<TaxAmount>"
7      And  user clicks calculate button
8      Then user gets "<TotalAmount>"
9
10     Examples:
11       | InvoiceAmount      | TaxAmount         | TotalAmount     |
12       | 100                | 50                | 150             |
13       | 200                | 10.5              | 210.5           |
```

Save the file and run it

```
Invoice.feature
1  Feature: Calculate invoice amount
2
3    Scenario Outline: Invoice amount - p
4      Given user is on invoice page
5      When user enters invoice amount "<
6      When user enters tax amount "<TaxA
7      And  user clicks calculate button
8      Then user gets "<TotalAmount>"
9
10     Examples:
11       | InvoiceAmount      | TaxAmount
12       | 100                | 50
13       | 200                | 10.5
```

Undo Typing	Ctrl+Z
Revert File	
Save	Ctrl+S
Open With	›
Show In	Alt+Shift+W ›
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Quick Fix	Ctrl+1
Shift Right	
Shift Left	
Pretty Format	Ctrl+Shift+F
Find Step	F3
Recalculate steps	F5
Coverage As	›
Run As	›
1 Cucumber Feature	

Copy the step definition code

Console

<terminated> Invoice.feature [Cucumber Feature] C:\Program Files\Java\jdk1.8.0_191\bin\javaw.exe (09-Mar-2021, 3:25:33 PM)

```
@Given("user is on invoice page")
public void user_is_on_invoice_page() {
    // Write code here that turns the phrase above into concrete actions
    throw new io.cucumber.java.PendingException
}

@When("user enters invoice amount {string}")
public void user_enters_invoice_amount(String s
    // Write code here that turns the phrase ab
    throw new io.cucumber.java.PendingException
}

@When("user enters tax amount {string}")
public void user_enters_tax_amount(String strin
    // Write code here that turns the phrase ab
    throw new io.cucumber.java.PendingException
}

@When("user clicks calculate button")
public void user_clicks_calculate_button() {
    // Write code here that turns the phrase above into concrete actions
    throw new io.cucumber.java.PendingException();
}

@Then("user gets {string}")
public void user_gets(String string) {
```

Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Select All	Ctrl+A
Find/Replace...	
Open Link	
Clear	
Remove All Terminated	
Terminate/Disconnect All	
Scroll Lock	
Word Wrap	
Preferences...	

Create step def file and paste the code. Import the packages and remove the exceptions

Invoice.java

CucumberMavenProject/src/test/java/StepDefinitions/Invoice.java

```java
public class Invoice {
    @Given("user is on invoice page")
    public void user_is_on_invoice_page() {

    }

    @When("user enters invoice amount {string}")
    public void user_enters_invoice_amount(String string)

    }

    @When("user enters tax amount {string}")
    public void user_enters_tax_amount(String string) {

    }

    @When("user clicks calculate button")
    public void user_clicks_calculate_button() {

    }

    @Then("user gets {string}")
    public void user_gets(String string) {
```

Save the file

Re-run the feature file

```
Invoice.feature
1  Feature: Calculate invoice amount
2
3    Scenario Outline: Invoice amount - p
4      Given user is on invoice page
5      When user enters invoice amount "<
6      When user enters tax amount "<TaxA
7      And  user clicks calculate button
8      Then user gets "<TotalAmount>"
9
10     Examples:
11       | InvoiceAmount      | TaxAmount
12       | 100                | 50
13       | 200                | 10.5
```

```
Undo Typing                Ctrl+Z
Revert File
Save                       Ctrl+S
Open With                  >
Show In                    Alt+Shift+W >
Cut                        Ctrl+X
Copy                       Ctrl+C
Paste                      Ctrl+V
Quick Fix                  Ctrl+1
Shift Right
Shift Left
Pretty Format              Ctrl+Shift+F
Find Step                  F3
Recalculate steps          F5
Coverage As                >
Run As                     >   1 Cucumber Feature
```

Notice the console o/p. The scenario got executed twice as expected and 10 steps got passed

```
Console
<terminated> Invoice.feature [Cucumber Feature] C:\Program Files\Java\jdk1.8.0_191\bin\javaw.exe (09-Ma
Scenario Outline: Invoice amount - possible combinations
  Given user is on invoice page
  When user enters invoice amount "100"
  When user enters tax amount "50"
  And user clicks calculate button
  Then user gets "150"

Scenario Outline: Invoice amount - possible combinations
  Given user is on invoice page
  When user enters invoice amount "200"
  When user enters tax amount "10.5"
  And user clicks calculate button
  Then user gets "210.5"

2 Scenarios (2 passed)
10 Steps (10 passed)
```

So this is how we use the ‘Scenario Outline’ and ‘Examples’ keyword combination to execute data driven testing in cucumber.

Thank you for reading!