BDD-> Behaviour driven Development

Feature:- Feature is file where we use gherkin keyword in order to write the scaneraio and file extension should be .feature

Stepdefinition: Here we are providing the business logic for sceanrios src/test/java/loginstep.java

Runner class:- here we map the feature file sceanrios with step definition and by using some cucumber options like glue and feature

Glue: In glue we are going specify the path of step definition

Feature: here we mentionthe path of feature file

Cucumber dependencies

<dependency>

<groupId>io.cucumber</groupId>

<artifactId>cucumber-core</artifactId>

<version>6.9.1</version>

</dependency>

<!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-testng -->

<dependency>

<groupId>io.cucumber</groupId>

<artifactId>cucumber-testng</artifactId>

<version>6.9.1</version>

</dependency>

<!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->

<dependency>

<groupId>io.cucumber</groupId>

<artifactId>cucumber-java</artifactId>

<version>6.9.1</version>

</dependency>

What is dryRun(boolean type)?

It is cumber option it will check the feature file scenario have all the step definition and if step definition is missing it will provide empty step definition

What is the difference between scenario and scenario outline?

When we want to execute one end2end scenario we go for scenario

And if we want to execute same script with multiple set of data we go for scenario outline mandatorily we use example keyword and we can achieve data driven testing

Scenario Outline: I want to login to application with multiple invalid credentials

Given I want to launch the browser

And I enter the url "http://localhost:8888/"

When login page is diplayed enter "<username>" and "<password>" and click on login button

Then Home page should be not be displayed and error messege should be displayed

Examples:

| username | password |

| admin | admin123 |

| admin123 | manager |

Scenario: I want to login to vtiger application using invalid credentials

Given I want to launch the browser

And I enter the url "http://localhost:8888/"

When Login page is displayed enter the username "admin123" and password "manager123" and click on login

Then Home page should be not be displayed and error messege should be displayed

**What is monochrome?**

In order to read neat console output we use monochrome option

**What is plugin option?**

To get scenario steps in console and also we use it for report generation

**How will you achieve data driven without using scenario outline?**

**In scenario after the step we mention the datatable and in step definition we are going to convert into lists or maps**

Feature: DataDriven

Scenario: Read the data using list

When user want to fill the registration form

|user|user@gmail.com|user123|12345|

Scenario: read the data using map

When user want to fill the registration form in application

|username|email|password|phNo|

[|user|user@gmail.com|user123|12345|](mailto:|user|user@gmail.com|user123|12345|)

**public** **class** DataDrivenStepdef {

@When("user want to fill the registration form")

**public** **void** user\_want\_to\_fill\_the\_registration\_form(DataTable dataTable) {

List<List<Object>> data = dataTable.asLists(Object.**class**);

System.***out***.println(data.get(0).get(0));

System.***out***.println(data.get(0).get(1));

System.***out***.println(data.get(0).get(2));

System.***out***.println(data.get(0).get(3));

}

@When("user want to fill the registration form in application")

**public** **void** user\_want\_to\_fill\_the\_registration\_form\_in\_application(DataTable dataTable) {

List<Map<Object, Object>> data = dataTable.asMaps(Object.**class**, Object.**class**);

System.***out***.println(data.get(0).get("username"));

System.***out***.println(data.get(0).get("email"));

System.***out***.println(data.get(0).get("password"));

System.***out***.println(data.get(0).get("phNo"));

}

}

**What is Background?**

**The common scenario steps we keep it in background and it will map for every scenario or execute for every scenario**

Feature: Login feature

Background:

Given I want to launch the browser

And I enter the url "http://localhost:8888/"

Scenario: I want to login to vtiger application using valid credentials

When Login page is displayed enter the username and password and click on login

Then Home page should be displayed

Scenario: I want to login to vtiger application using invalid credentials

When Login page is displayed enter the username "admin123" and password "manager123" and click on login

Then Home page should be not be displayed and error messege should be displayed

What are Hook in cucumber?

In order to execute pre and post condition we are going to use Hook and it contains only 2 annotations @Before and @After

**public** **class** Hook {

@Before

**public** **void** setUp() {

System.***out***.println("precondition");

}

@After

**public** **void** tearDown() {

System.***out***.println("post condition");

}

}

**How will you order the execution of Hook?**

Provide order option in @Before and @after and @Before execution order is ascending order and @After execution is in descending order

**How will you perform batch execution?**

With the help of tags ex:@SmokeTest

Feature: Login feature

Background:

Given I want to launch the browser

And I enter the url "http://localhost:8888/"

@SmokeTest

Scenario: I want to login to vtiger application using valid credentials

When Login page is displayed enter the username and password and click on login

Then Home page should be displayed

@RegressionTest

Scenario: I want to login to vtiger application using invalid credentials

When Login page is displayed enter the username "admin123" and password "manager123" and click on login

Then Home page should be not be displayed and error messege should be displayed

@Test

@CucumberOptions(features = {".\\src\\test\\resources\\Features"},

glue= {"stepDefinitions","hookClass"},

dryRun = **true**,

plugin = {"pretty"},

monochrome = **true**,

tags="@SmokeTest"

)

**public** **class** Runner **extends** AbstractTestNGCucumberTests{

}