Cucumber

By using cucumber we are able to provide gherkin language. What is the gherkin language in this gherkin language everyone able to understand that is the plane English. Why we need to use “cucumber” already “TestNg” there? In real time so many customer are there Non IT background How they will understand the code ? If u want to provide any code those people also understand or not we need to cross check.

How we provide the GHerkin language in Cucumber ?

The Cucumber completely depends on 3 classes

Feature:

In the feature class we able to provide

1. **Feature:** The feature will explain what we need to do.
   Ex: Salesforce login verification successful.
2. **Background:** if we want to do the Salesforce login verification successful. I need to provide Background.
   Ex: LaunchAUT under background.
   In Cucumber every Scenario before background will be execute
3. **Scenario:** The Scenario will execute only one time
4. **Scenarios outline:** The scenario outline will execute no.of time based on the example.
5. **Gherkin:**
   Given : Launch the Application URL
   When : user provide uname and pass
   Then : Click on login button
   And : get the title.
6. Examples

StepDefination: in StepDefination what ever we provide in side of the feature file the step related code we need to provide under StepDefination.

TestRunner: If you want to execute any Cucumber code always we need to execute inside of the TestRunner. In testrunner we are going to provide some options.

@Cucumber Class
@CucumberOption
feature=”feature file location”; (feature location)

glue=”StepDefination package name”; inside of the glue we provide StepDefination package name;

MonoChrome=true; MonoCrome is the readable format we are able to get the readable format inside of the console.

dryrun=true; when you guys are provide the dryrun will compare feature file and StepDefination. What ever you provide inside of the feature file the feature file related information is there or not in StepDefination the dryrun will verify. If the miss any class in Gherkin then we will be get error.

stricky = true; if u guys are skip any feature file stricky will be handle StepDefination.

tags={"Retesing"} if we want to do any type of testing we need to provide tags. like "Retesting","Regression Testing"….

Format={PrettyPrint"} format.
This is we called Cucumber.

How to do the Cucumber setup?
We need some "dependencies"

1. Cucumber plugin
2. Natural
3. Selenium with java
4. Juunit
5. TestNg
6. Cucumber java
7. Cucumber JVM
8. Gherkin
9. Cobetura

These are all required in Cucumber project.

1. open the "Eclipse IDE" click on "help" goto "Marketplace" search for the "Cucumber plugin" click on go click on install.
2. After installation we need to get "Naturals" again go help click on "Eclipse Marketpalce" and search for the "Natural plugin" if not available close the Marketplace.
3. Go to the Chrome and search for the "natural eclipse" and go to stackflow copy that link ("https://rlogiacco.github.io/Natural") and paste in "install new software".
4. After completion of the Setup create "Maven" project with name "ABS9_SCucumber".

Open the “pom.xml” and go to maven repository search for the “selenium java dependency”, “TesNg dependency”, “Junit dependency”, “Cucumber JVM” dependency and “Gherkin dependency”.

We need to create “Feature file” and always we need to create “Feature Folder” inside of the “src/test/resources”.

1. Click on the right button on the “src/test/resources” create a Folder with “Feature” name.
2. Inside of the Folder we need to be create file. Click on the right button on the folder select “File” create file with “login_sales.feature” (here “feature” is extension must be mention).
3. Now setup the “StepDefination” and “Testrunner”.
4. In the “src/test/java” click on the right button we need to goto “new” and create a “package” with name StepDefination. once u guys are complete created package click on the “Finish”.
5. Create the 2 Java classes. Create the 1st class with “StepDefination” class name.
6. Create a 2nd Class with “TestRunner” class name.
7. If we want to write a code first we have to write code in “login_sale.feature”. in this feature we able to create “Scenario”, “Gherkin” language. First of all we have to write

Feature: login the expected application

Scenario: Verify Salesforce login

After that we need to provide “Gherkin” language.

Given launch the salesforce application
When user need to enter uname and pwd
Then click on Salesforce login

After that we need to providing “what we are providing inside of the feature file as tease that related information we need to map inside of the StepDefination. In the StepDefination we need to copy the “Structure”.

In the “StepDefination” class we have to be write the entire code:

```
package Step_Defination;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.chrome.ChromeDriver;

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

public class StepDefination {
    WebDriver driver = new ChromeDriver();
    @Given("launch the salesforce application")
    public void launch_the_salesforce_application() {
          driver.manage().window().maximize();
          driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);
         driver.get("");
    }

    @When("user need to enter uname and pwd")
    public void user_need_to_enter_uname_and_pwd() throws Exception
    {

driver.findElement(By.id("username")).sendKeys("SyedRizwanTesting@gmail.com");
         Thread.sleep(2000);

driver.findElement(By.id("password")).sendKeys("SyedRizwanTesting@gmail.com");
         Thread.sleep(2000);
    }

    @Then("click on Salesforce login")
    public void click_on_salesforce_login()
    {
        driver.findElement(By.id("Login")).click();
    }

}
```

Next go to “TestRunner” Class and we need to enter all below code

```
package Step_Defination;
import org.junit.runner.RunWith;
import io.cucumber.junit.Cucumber;
import io.cucumber.junit.CucumberOptions;
@RunWith(Cucumber.class)
@CucumberOptions()
public class TestRunner {
}
```

If we want to do the Cucumber Project we need these 3 files mandatory (feature file, StepDefination file, TestRunner file).

In “TestRunner” class we need to write all this below code

```
package Step_Defination;
import org.junit.runner.RunWith;
import io.cucumber.junit.Cucumber;
import io.cucumber.junit.CucumberOptions;

@RunWith(Cucumber.class)
@CucumberOptions(

  features="C:\\Users\\HP\\Desktop\\Eclipse\\ABS9_SCucumber\\src\\test\\resources\\Feature\\login_sale.feature",
      glue = {"Step_Defination"}      //here we need to provide Step_Defination package name

      )

public class TestRunner {
}
```

Amazon search: Create the “amazonSearch.feature” file in “feature Folder” and in “StepDefination” we have to copy just two method like

```
@When("Enter the data in seacrh bar")
public void enter_the_data_in_seacrh_bar() {
// Write code here that turns the phrase above into concrete actions
throw new io.cucumber.java.PendingException();
}

@Then("Click on searchbutton in amazon")
public void click_on_searchbutton_in_amazon() {
// Write code here that turns the phrase above into concrete actions
throw new io.cucumber.java.PendingException();
}
```

And after that we have to remove the inside of the lines and we have to mention code like this.

```
@When("Enter the data in seacrh bar")
public void enter_the_data_in_seacrh_bar() throws Exception {
driver.findElement(By.id("")).sendKeys("Java with Selenium");
Thread.sleep(2000);
}
```

```
@Then("Click on searchbutton in amazon")
public void click_on_searchbutton_in_amazon()
{
driver.findElement(By.id("")).click();
}
```

After complete this code we have to write these 2 methods

```
static By s_bar=By.id("");
static By Icon=By.id("");  at above of the the
@Given("Launch the Application").
```

Now again come to here

```
@When("Enter the data in seacrh bar")
public void enter_the_data_in_seacrh_bar() throws Exception {
driver.findElement(By.id("")).sendKeys("Java with Selenium");
Thread.sleep(2000);
}

@Then("Click on searchbutton in amazon")
public void click_on_searchbutton_in_amazon()
{
driver.findElement(By.id("")).click();
}
```

Now we have to write code like this

```
//Amazon
@When("Enter the data in seacrh bar")
public void enter_the_data_in_seacrh_bar() throws
Exception {
driver.findElement(s_bar).sendKeys("Selenium");
Thread.sleep(2000);
}

@Then("Click on searchbutton in amazon")
public void click_on_searchbutton_in_amazon()
{
driver.findElement(Icon).click();
}
```

we need to go to "Testrunner" file we need to change "feature" value

Featurre="C:\Users\HP\Desktop\Eclipse\ABS9_SCucumber\src\test\resources\Feature\amazonSeacrh.feature",

```
package Step_Defination;

import org.junit.runner.RunWith;
```

```
import io.cucumber.junit.Cucumber;
import io.cucumber.junit.CucumberOptions;

@RunWith(Cucumber.class)
@CucumberOptions(

  //features="C:\\Users\\HP\\Desktop\\Eclipse\\ABS9_SCucumber\\src\\test\\resources\\Feature\\login_sale.feature",

  features="C:\\Users\\HP\\Desktop\\Eclipse\\ABS9_SCucumber\\src\\test\\resources\\Feature\\amazonSeacrh.feature",
      glue = {"Step_Defination"}      //here we need to provide Step_Defination package name

      )

public class TestRunner {
}
```

Today we are going to discuss about

- How to use the background
- How to use the Scenario outline
- How to use the example

In the feature folder we need to create a file with name “amazonValidate.feature”.

```
Feature: Verify the amazon
Background:
Given launch the Application

Scenario: Verify mobile
Given Validate the Mobile button

Scenario: Verify Sell
Given Validate the Sell button

Scenario: Verify search bar
Given Enter the data in seacrh bar
When Click on searchbutton in amazon
```

Every “Scenario” before “Background” will be execute.

Just click on right button click on “Run as” click on “Cucumber Feature” and we need to take undefined methods copy 3 methods and go to “StepDefination” and paste here.

Next below of these 2 statements

```
static By s_bar=By.id("twotabsearchtextbox");
static By Icon=By.id("nav-search-submit-button");
```

we need to write these below statements

```
static By mobile = By.xpath("//a[text()='Mobiles']");
static By sell = By.xpath("//a[text()='Sell']");
```

In “StepDefination” we need to write code like this

```
package Step_Defination;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

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

public class StepDefination {
  WebDriver driver = new ChromeDriver();
  //@Given("launch the salesforce application")


  static By s_bar=By.id("twotabsearchtextbox");
  static By Icon=By.id("nav-search-submit-button");
  static By mobile = By.xpath("//a[text()='Mobiles']");
  static By sell = By.xpath("//a[text()='Sell']");
  @Given("Launch the Application")


  public void launch_the_salesforce_application() {
      driver.manage().window().maximize();

driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS)
;
     //driver.get("https://login.salesforce.com/?locale=in");
      driver.get("https://www.amazon.in/");
```

```
  }

  @When("user need to enter uname and pwd")
  public    void    user_need_to_enter_uname_and_pwd()    throws
Exception
  {

driver.findElement(By.id("username")).sendKeys("SyedRizwanTest
ing@gmail.com");
     Thread.sleep(2000);

driver.findElement(By.id("password")).sendKeys("SyedRizwanTest
ing@gmail.com");
     Thread.sleep(2000);
  }

  @Then("click on Salesforce login")
  public      void      click_on_salesforce_login()      throws
InterruptedException
  {
    driver.findElement(By.id("Login")).click();

  }

  //Amazon
  @When("Enter the data in seacrh bar")
  public void enter_the_data_in_seacrh_bar() throws Exception {
      driver.findElement(s_bar).sendKeys("Selenium");
      Thread.sleep(2000);
  }

  @Then("Click on searchbutton in amazon")
  public    void    click_on_searchbutton_in_amazon()    throws
InterruptedException
  {
    driver.findElement(Icon).click();
    Thread.sleep(2000);
    //driver.quit();
  }

  @Given("Validate the Mobile button")
  public void validate_the_mobile_button()
  {
```

```
    driver.findElement(mobile).click();
  }

  @Given("Validate the Sell button")
  public void validate_the_sell_button()
  {
    driver.findElement(sell).click();
  }
```

After this above code we have to go “TestRunner” file copypath Of the file in feature. And run this program.
After Execution we can write in code in ‘amazonValidate.feature’ file like this.

```
Feature: Verify the amazon
Background:
Given Launch the Application

Scenario: Verify mobile
Given Validate the Mobile button
When Close Active window

Scenario: Verify Sell
Given Validate the Sell button
When Close Active window

Scenario: Verify search bar
Given Enter the data in seacrh bar
When Click on searchbutton in amazon
Then Close Active window
```

After run this Cucumber feature we need to take this below method

```
@Then("Close Active window")
public void close_active_window() {
driver.close();
}
```

And update in “StepDefination” file. As a below

```
@Then("Close Active window")
public void close_active_window() {
driver.close();
} again come to “TestRunner” execute the program.
```

Scenario Outline: Create a new file with name “sales_f.feature” in “Feature” folder. In this file we need to write the code like this

```
Feature: Verify and Valid the salesforce
Background:
Given Launch the Application
Scenario Outline: Validate the Expected
Given user enter <username> and <password>
Then click on Salesforce login

Examples:
|username |password|
|Syed123@gmail.com| Syedrfhia|
|sayyad342@yahoo.com|Rizzu453|
|Rizzu432| Rizwan657|
```

After we have to go to “StepDefination” and we have to be write the code like this.

```
//Scenario Outline
@Given("^user enter(.*) and (.*)$")
public void user_enter_and(String username, String
password) throws Exception {
driver.findElement(By.id("username")).sendKeys(us
ername);
Thread.sleep(2000);
driver.findElement(By.id("password")).sendKeys(pa
ssword);
Thread.sleep(2000);
driver.findElement(By.id("Login")).click();
}
```

After that we have to copy the “file copypath” of the “sale_f.feature” and next come to “StepDefination” file and change the “url” as “https://login.salesforce.com/?locale=in”.

How to do the retesting, regression testing and Smoke Testing?

Create a new file in “Feature” folder with name “typesoftesting.feature”. in this file we need to write the code like this

```
Feature: Expected Testing
Background:

Given Launch the Application

Scenario: Verify mobile
Given Validate the Mobile button
When Close Active window

Scenario: Verify Sell
Given Validate the Sell button
When Close Active window

Scenario: Verify search bar
Given Enter the data in seacrh bar
When Click on searchbutton in amazon
Then Close Active window
```

Right now I want to test the “Retesting” and “SmokeTesting” you need to write the code like this.

```
Feature: Expected Testing
Background:

Given Launch the Application
@Retesting
Scenario: Verify mobile
Given Validate the Mobile button
When Close Active window
@Smoketesting
Scenario: Verify Sell
Given Validate the Sell button
When Close Active window

Scenario: Verify search bar
Given Enter the data in seacrh bar
When Click on searchbutton in amazon
Then Close Active window
```

And next go to file and copy the “typeoftestng.feature” filepath. And paste in “TestRunner” file. Next we need to write the tags=”@Retesting” in TestRunner file.

package Step_Defination;

```
import org.junit.runner.RunWith;

import io.cucumber.junit.Cucumber;
import io.cucumber.junit.CucumberOptions;

@RunWith(Cucumber.class)
@CucumberOptions(

features="C:\\Users\\HP\\Desktop\\Eclipse\\ABS9_SCucumber\\src\\test\\resources\\Feature\\typesoftesting.feature",

glue = {"Step_Defination"},     //here we need to provide Step_Defination package name
tags= "@Retesting"

	)

public class TestRunner {
}
```

If we want to change the url go to Amazon copy the url and oaste in “StepDefination”.