

CUCUMBER

CUCUMBER FRAMEWORK

1. BDD (Behaviour Driven Development) with the execution of TDD (Test Driven Development).
2. Used to convert the exact requirements into **plain English language**.
3. It support insprint automation.
4. Non-Technical people can also understand the automation coverage.
5. Cucumber supports **Ruby,Java,Grovy,Python,.NET and PHP**.
6. We can use the **Gherkin language or Gherkin Keyword** (Plain English).
7. Cucumber options.
8. Cucumber Reports.

CUCUMBER ANNOTATIONS

- @Given
- @When
- @Then
- @CucumberOptions

BDD FRAMEWORK

1. Cucumber
2. Jbehave
3. Nbehave
4. Specflow

GHERKIN Keyword / GHERKIN Language

- Feature
- Scenario
- Given
- When
- Then
- And
- But
- Scenario Outline
- Examples
- Background

CUCUMBER

Feature

1. Feature defines the logical test functionality test [What you will test the feature file].
2. **Feature** keyword is present at the starting of the feature file.

Scenario

1. Business rule[Provides the scenario name /purpose of the scenario].

Given

1. Some precondition step.

When

1. Some key actions.

Then

1. To observe the outcomes or validation.

Scenario Outline

1. Used to execute the same scenario with multiple times with different set of test datas.

Examples

1. Used to pass the parameters using the | **(pipe symbol)** | separate the each variables.

But

1. Used to add negative type comments.

Background

1. Define steps which are common to all the tests in the feature file.

And

1. Used to add conditions in feature file.

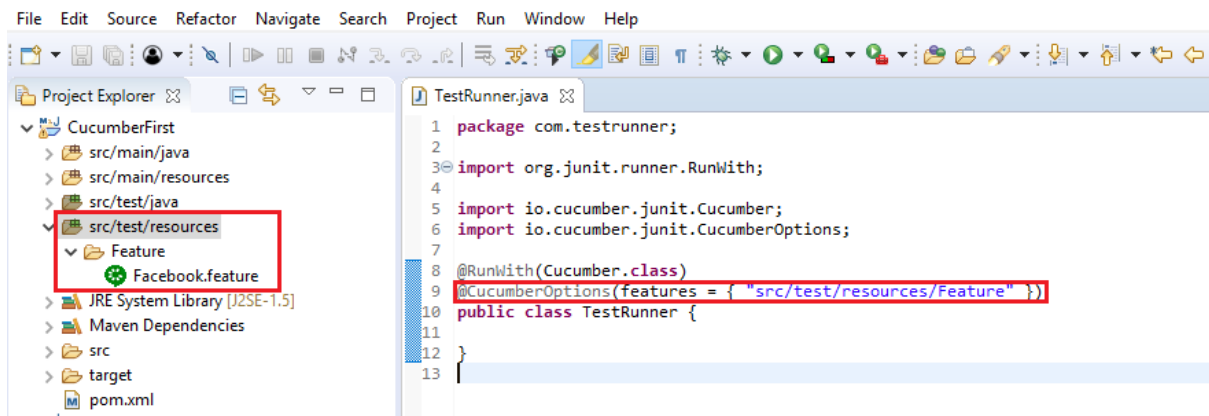
CUCUMBER

CUCUMBER OPTIONS

- features
- glue
- dryRun
- monochrome
- plugin
- pretty
- tags
- strict

features

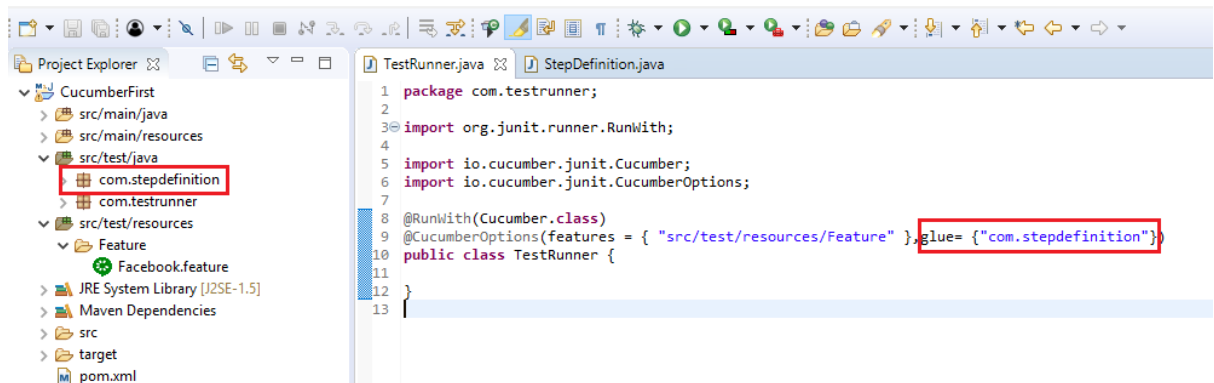
- ❖ **feature** option is used to locate the Feature folder in project folder structure.



FEATURE

Glue

- ❖ **glue** option is used to locate the stepDefinition class.

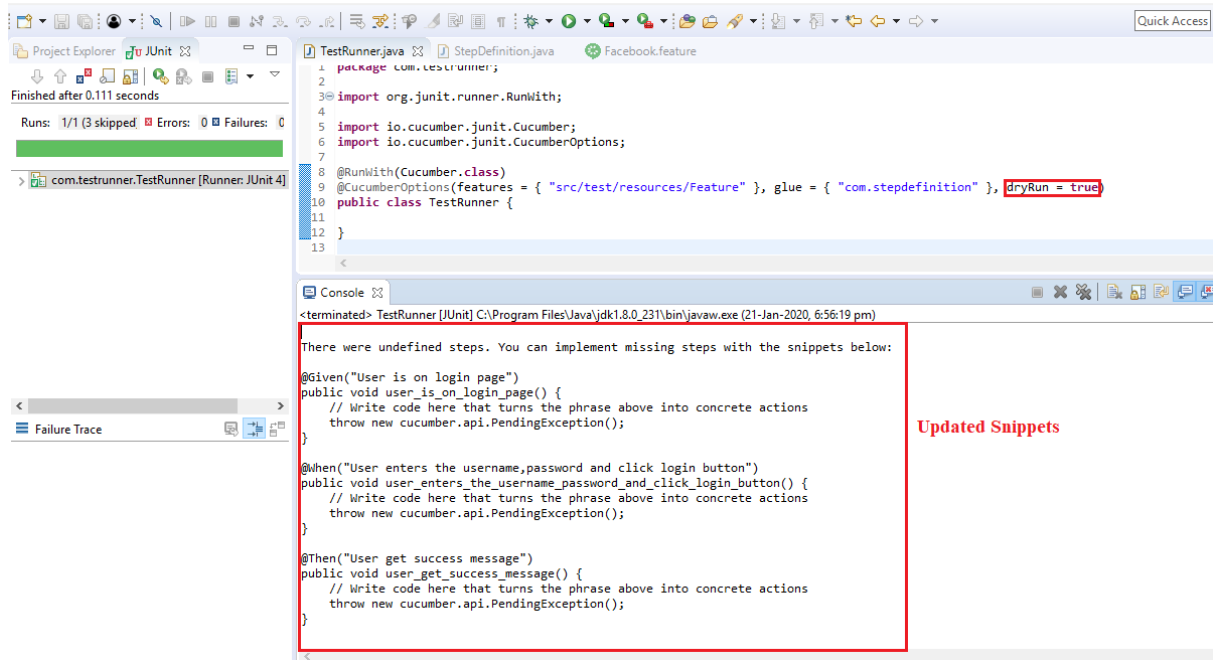


GLUE

CUCUMBER

dryRun

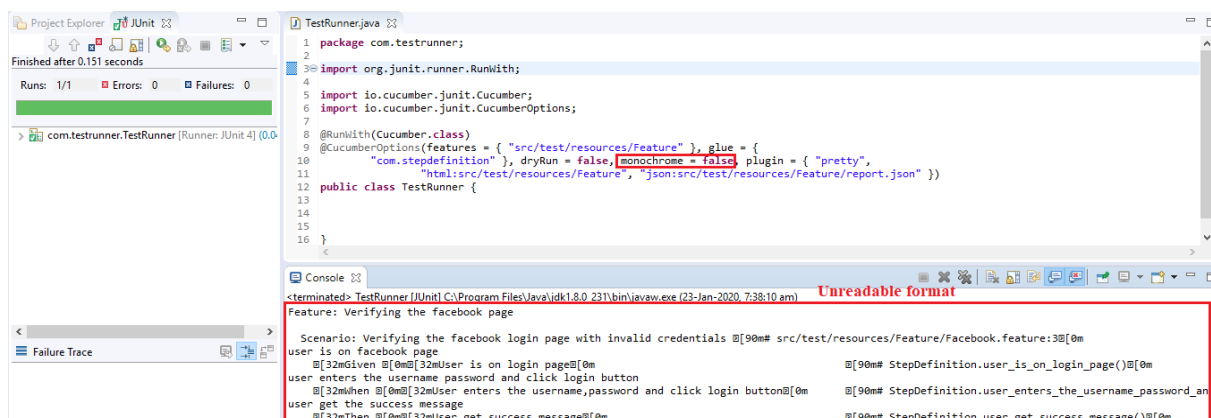
- ❖ **dryRun** option can either set as true or false.
- ❖ if it is set as **true** means check the every step mentioned in the feature file has corresponding code written in StepDefinition class or not. so in case any of the step is mismatch it update the snippets alone.



dryRun

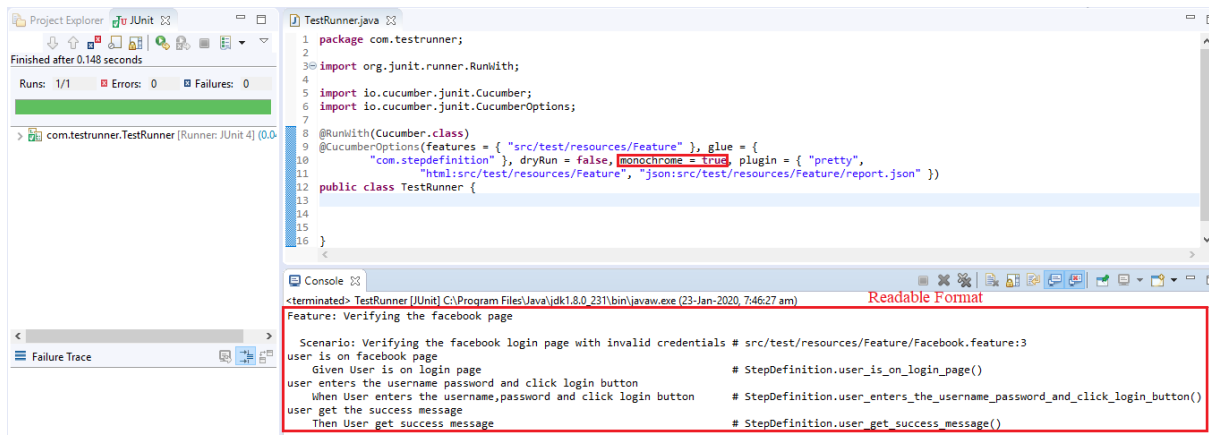
monochrome

- ❖ **monochrome** option can either set as true or false.
- ❖ If it is set as **true**, it means that the **console output** for the Cucumber test are much more readable.
- ❖ if it is set as **false**, then the **console output** is not as readable.



Monochrome(false)

CUCUMBER



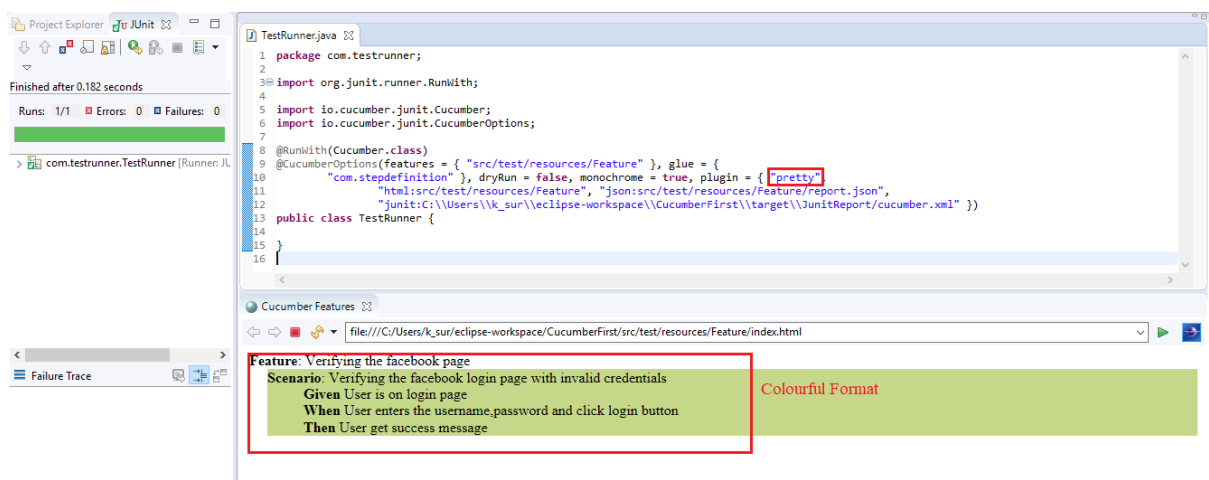
Monochrome(true)

plugin

- ❖ **plugin** option is used to specify different formatting options for the output reports(html,json,xml).

pretty

- ❖ **pretty** option is used to Print the **Gherkin** source with additional colors and stack traces for errors.



Pretty

tags

- ❖ **tags** option is for which tags in the feature file should be executed
- ❖ **tags**={“@<tag name>”}

strict

- ❖ **strict** option is **true** will fail execution, if there are undefined or pending steps

CUCUMBER

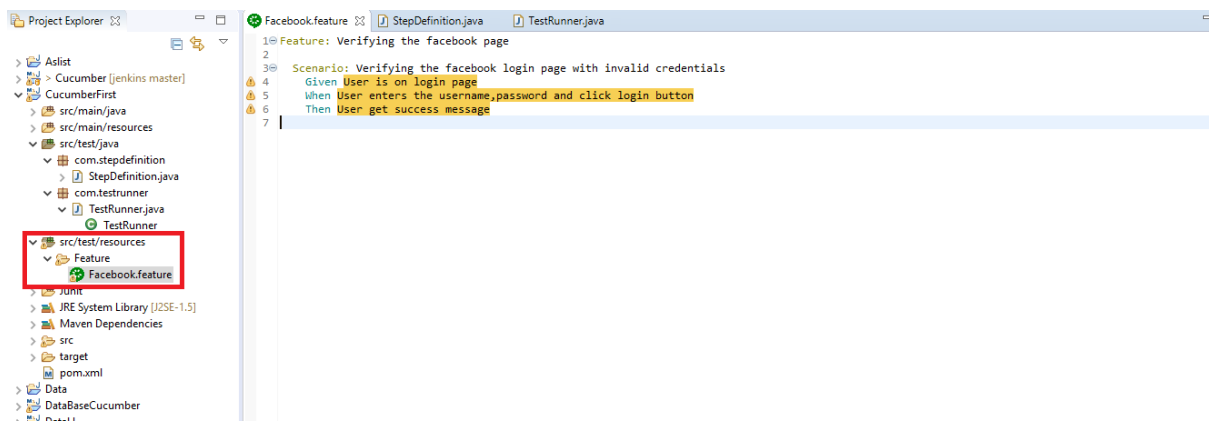
Cucumber Options

Option	Purpose	Default Value
features	set: The paths of the feature files	{}
Glue	set: The paths of the step definition files	{}
Tags	instruct: What tags in the feature files should be executed	{}
dryRun	true: Checks if all the steps have the Step Definition	false
monochrome	true: Display the console output in much readable format	false
plugin	set: What all the report formats to use	{}
Strict	true: Will fail execution, if there are undefined or pending steps	false

Steps:

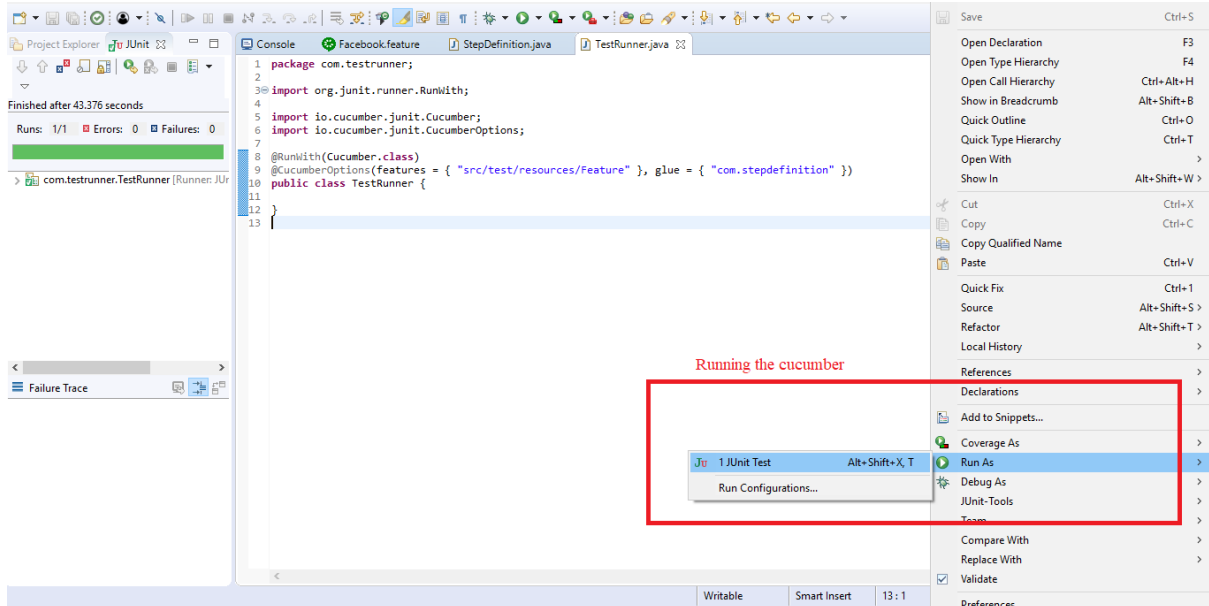
1. Add the cucumber plugin [cucumber eclipse plugin] in Eclipse marketplace.
2. Create maven project
3. Add the cucumber dependency(cucumber-junit,cucumber-java,selenium-java).
4. Create feature file in (src/test/resources).
5. Create TestRunner class(src/test/java).
6. Generate the snippets.
7. Copy the step definition files.
8. Copy the generated snippets and paste in the Stepdefinition class.
9. Run the code using testrunner class (right click → run as → junit)

1.CUCUMBER PROGRAM TO LAUNCH FACEBOOK

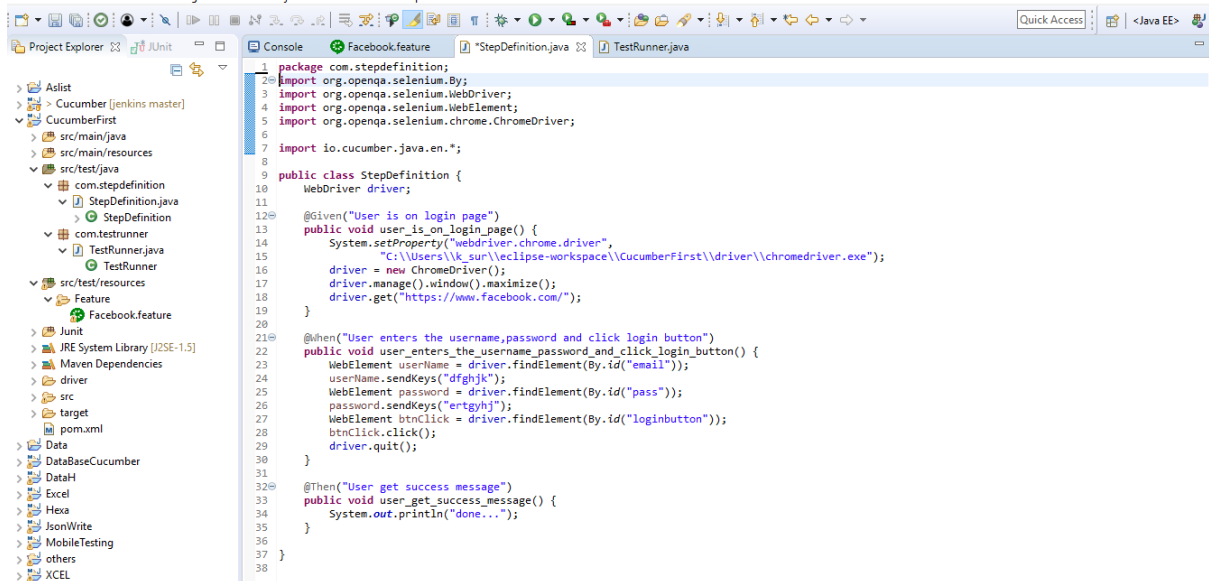


1.1 Feature File

CUCUMBER



1.2 TestRunnerClass

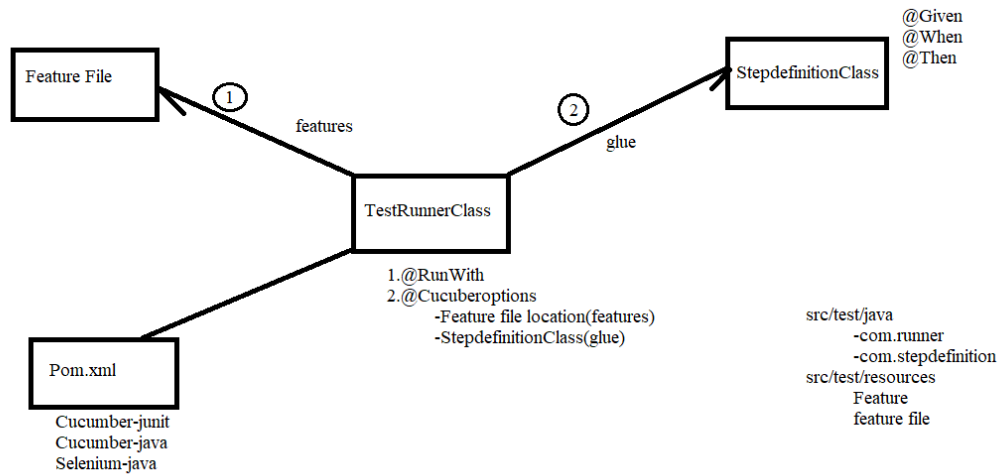


1.3 StepdefinitionClass

CUCUMBER

Gherkin Keywords

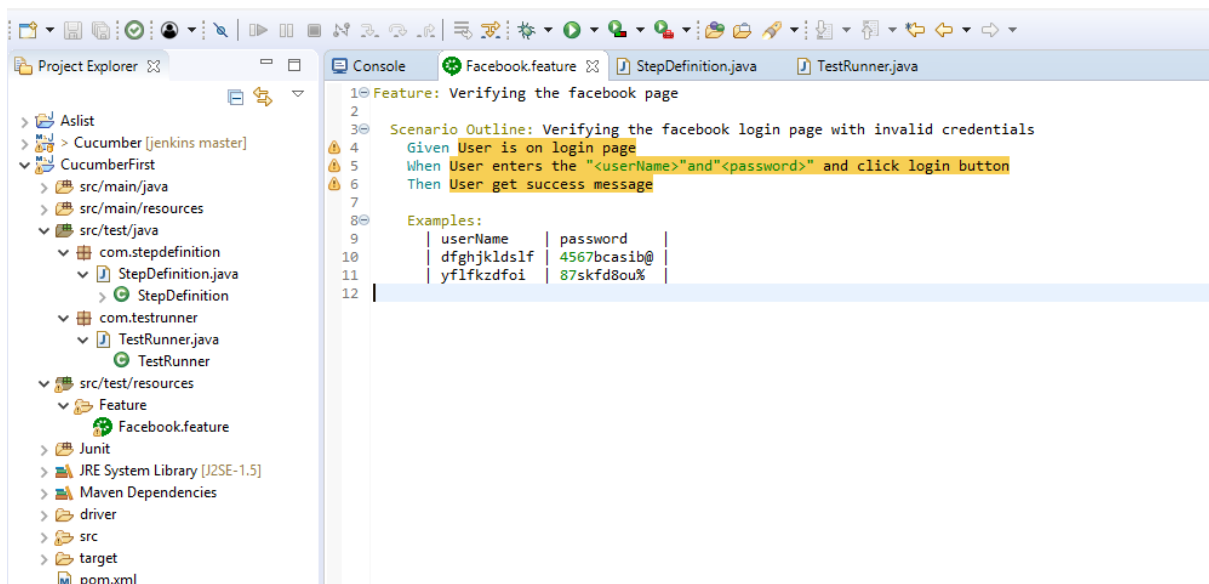
Feature
Scenario
Given
When
Then



1.4 Cucumber Architecture

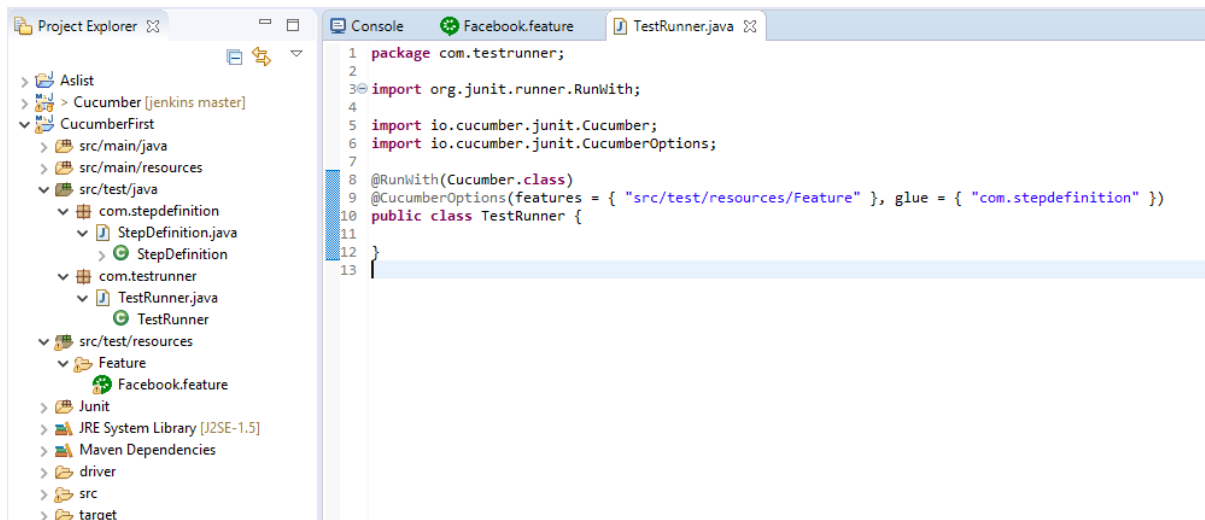
2.LAUNCH FACEBOOK AND PASSING THE PARAMETER USING EXAMPLE

- In cucumber using **Scenario Outline** and **Example** keyword we can pass multiple parameters for same scenario.

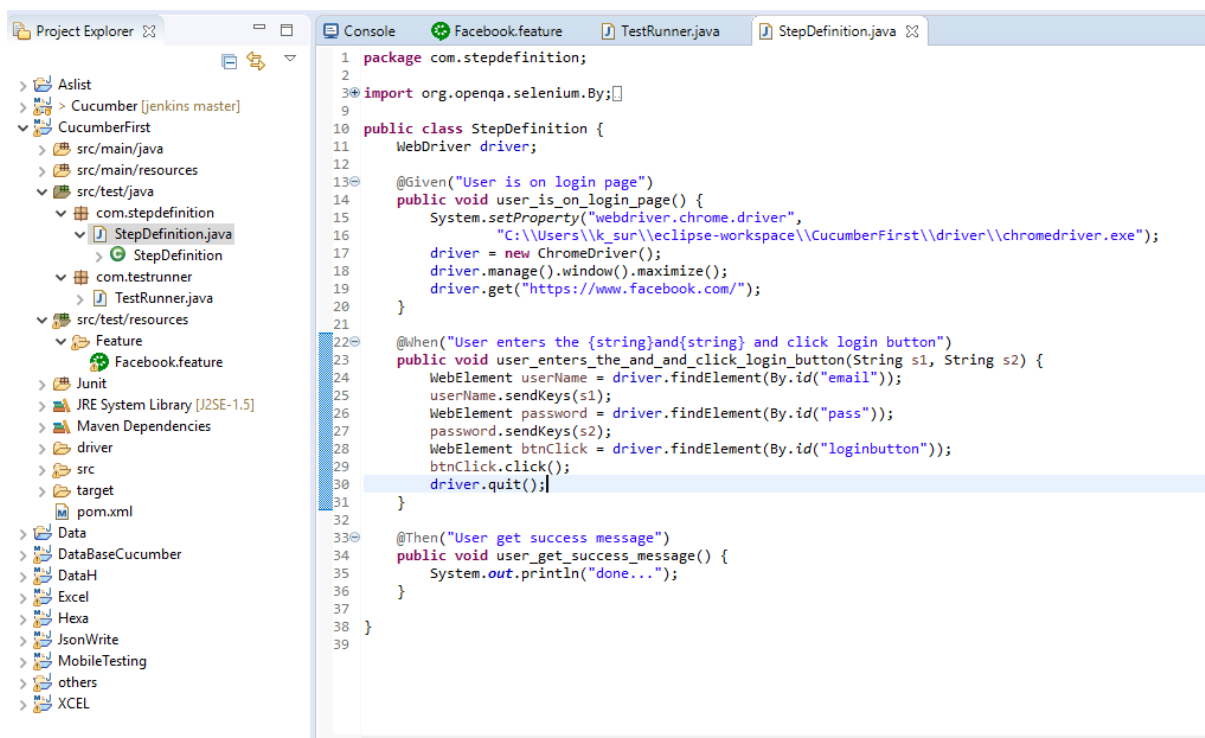


2.1 Feature file

CUCUMBER



2.2 TestRunnerClass



2.3 StepdefinitionClass

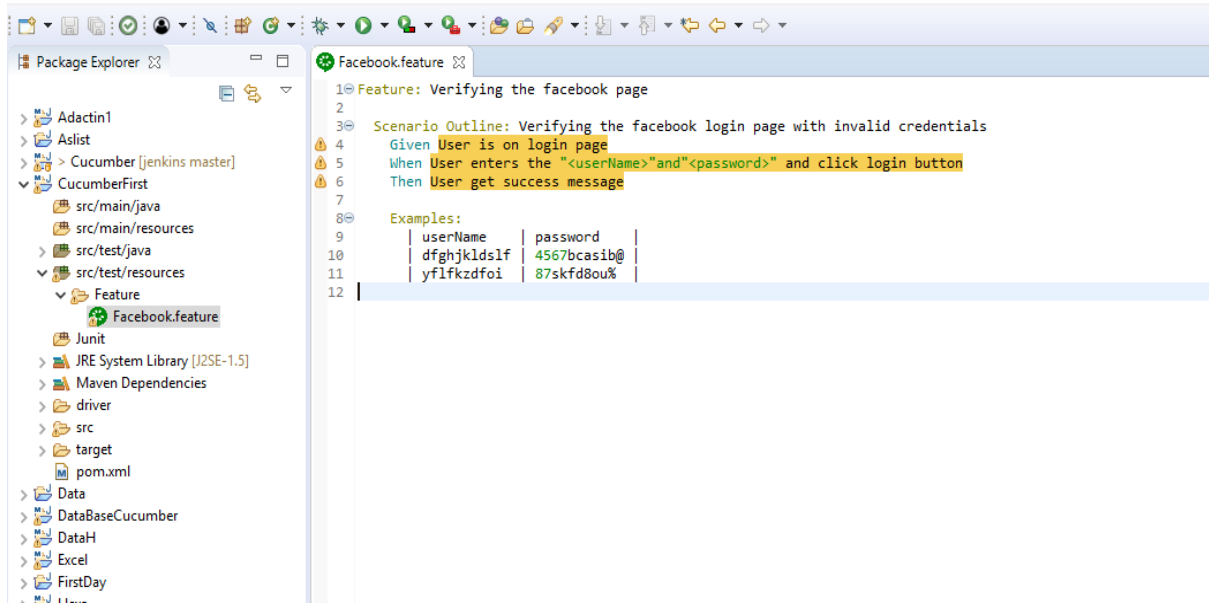
3.DESIGN PATTERNS

- Page Object Model(POM).
- Singleton.

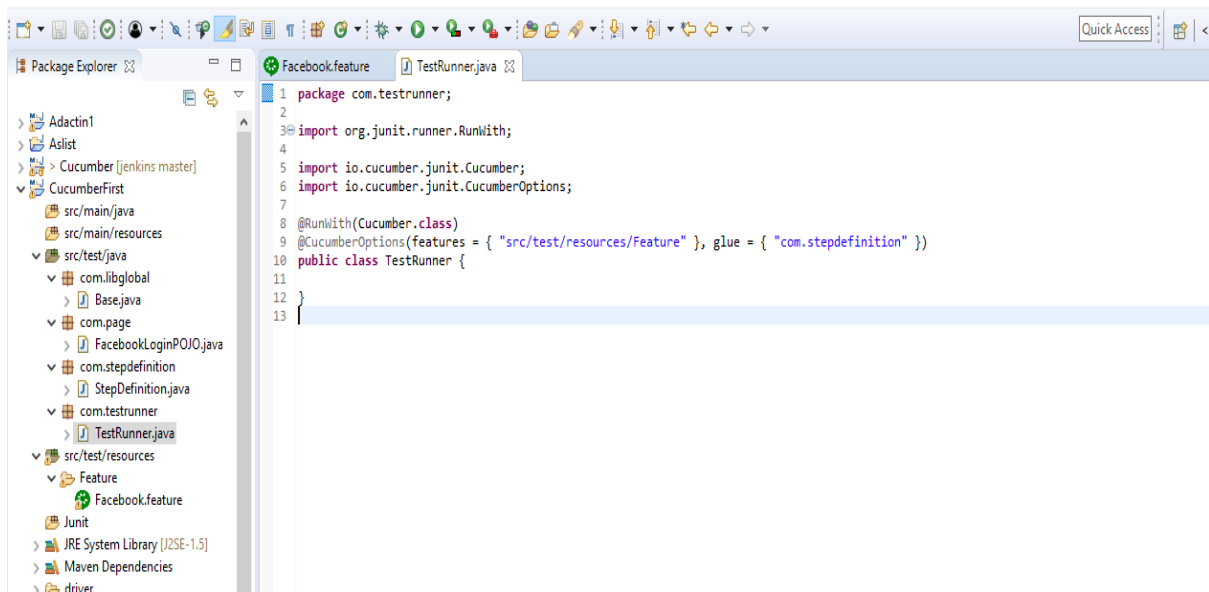
CUCUMBER

3.1 CUCUMBER WITH BASE CLASS AND POM

- ❖ It is widely used design pattern in **Cucumber** for enhancing test maintenance and reducing code duplication.

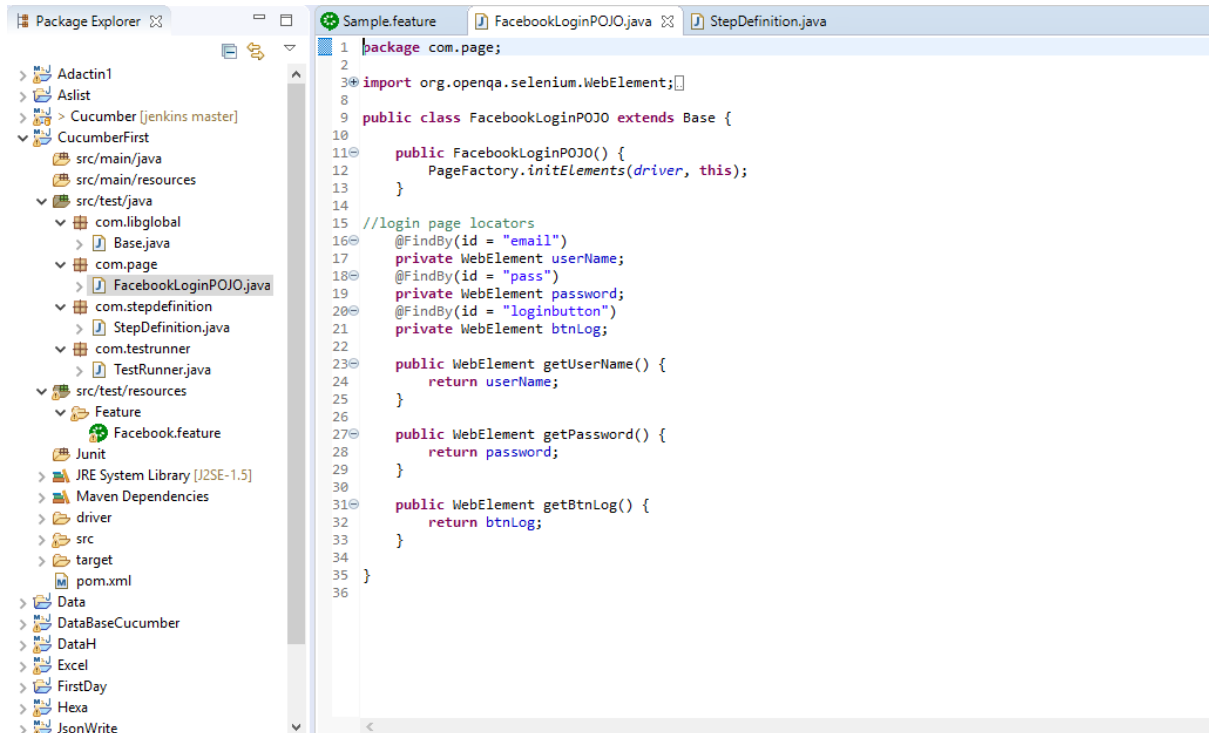


3.1.1 Feature file

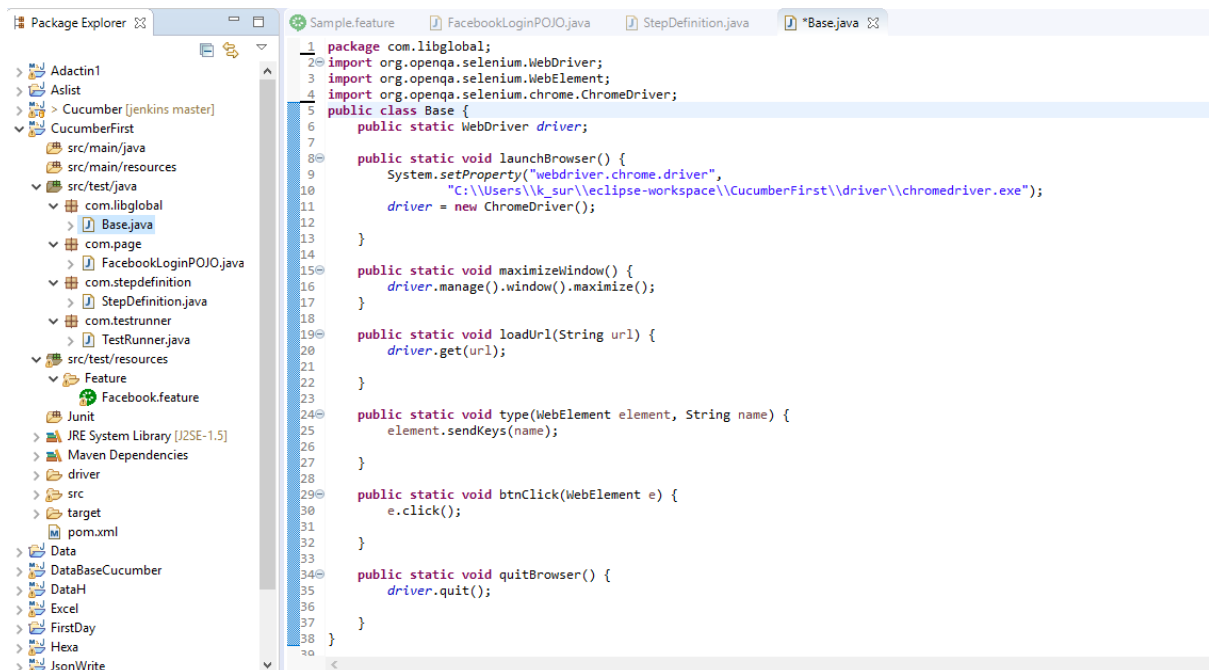


3.1.2 TestRunnerClass

CUCUMBER

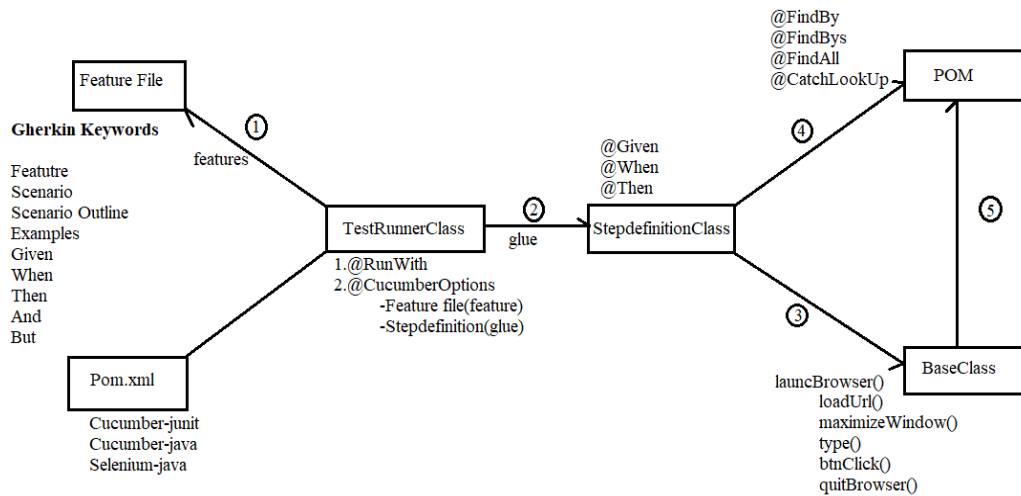


3.1.3 POJO CLASS



CUCUMBER

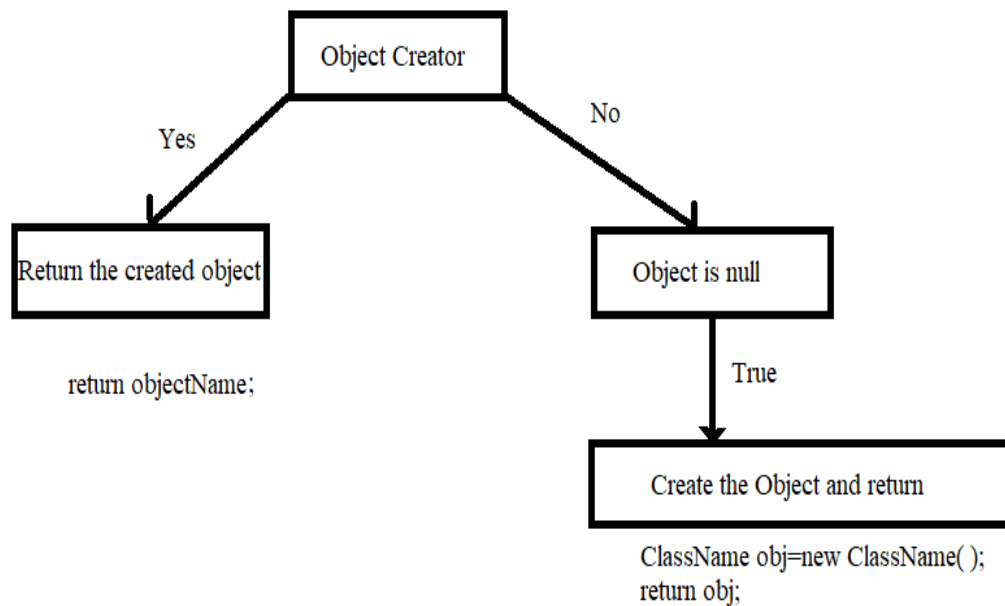
3.1.4 BaseClass



3.1.5 Architecture

3.2 CUCUMBER WITH SINGLETON

❖ Reducing the object creation we go for **singleton** design pattern.



3.2.1 SINGLETON

CUCUMBER

```
Adactin.feature
1 Feature: Verifying the adactin web page
2
3 Scenario Outline: Verify the login with valid credentials
4   Given User is on adactin page
5   When User enters "<username>" and "<password>"
6   And User clicks login button
7   Then User verify success message
8
9 Examples:
10  | username | password |
11  | nitishselvakumar | s1740034 |
12
13 Scenario Outline: Verify the Search Hotel page
14  When User select "<Location>" and "<Hotels>" and "<Room Type>" and "<Number of Rooms>" and "<Adults per Room>" and "<Children per Room>"
15  And User click search button
16
17 Examples:
18  | Location | Hotels | Room Type | Number of Rooms | Adults per Room | Children per Room |
19  | Sydney | Hotel Creek | Standard | 1 - One | 1 - One | 1 - One |
20
21 Scenario: verify the Select Hotel page
22  When User select the Hotel
23  And User click the continue button
24
25 Scenario Outline: verify the search hotel page
26  When User enter "<First Name>" and "<Last Name>" and "<Billing Address>" and "<Credit Card No>" and "<Credit Card Type>" and "<Select month>" and "<Select year>" and "<CVV Number>"
27  And User click BookNow button
28
29 Examples:
30  | First Name | Last Name | Billing Address | Credit Card No | Credit Card Type | Select month | Select year | CVV Number |
31  | nitish | selva | Chennai | 1234567890123456 | VISA | January | 2022 | 826 |
32
33 Scenario: verify the confirmation page
34  When User click search hotel
35
```

3.2.3 Feature File

```
Package Explorer  JUnit  TestRunner.java
└─ Adactin [Cucumber master]
   └─ src/test/java
      └─ com.libglobal
         └─ BaseClass.java
      └─ com.page
         └─ PageObjectManager.java
      └─ com.stepdefinition
         └─ StepDefinition.java
      └─ com.testrunner
         └─ TestRunner.java
      └─ org.pojo
         └─ BookHotelPOJOClass.java
         └─ ConfirmationPOJOClass.java
         └─ HotelConfirmPOJOClass.java
         └─ LoginPOJOClass.java
         └─ SearchHotelPOJOClass.java
      └─ src/test/resources
         └─ Features
            └─ Adactin.feature

1 package com.testrunner;
2
3 import org.junit.runner.RunWith;
4
5 import io.cucumber.junit.Cucumber;
6 import io.cucumber.junit.CucumberOptions;
7
8 @RunWith(Cucumber.class)
9 @CucumberOptions(features = { "src/test/resources/Features" }, glue = { "com.stepdefinition" })
10
11 public class TestRunner {
12
13 }
14
```

3.2.4 TestRunnerClass

CUCUMBER

```
package com.libglobal;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
public class BaseClass {
    public static WebDriver driver;

    public static void launchBrowser() {
        System.setProperty("webdriver.chrome.driver",
"C:\\\\Users\\k_sur\\Desktop\\Nitish\\nitish\\Cucumber\\driver\\chromedriver.exe");
        driver = new ChromeDriver();

    }

    public static void maximizeWindow() {
        driver.manage().window().maximize();
    }

    public static void loadUrl(String url) {
        driver.get(url);
    }

    public static void type(WebElement e, String value) {
        e.sendKeys(value);
    }

    public static void btnClick(WebElement e) {
        e.click();
    }

    public static void getTittle() {
        String title = driver.getTitle();
        System.out.println(title);
    }

    public static void getUrl() {
        String currentUrl = driver.getCurrentUrl();
        System.out.println(currentUrl);
    }

    public static void quitBrowser() {
        driver.quit();
    }

    public static void selectByVisibleText(WebElement element, String text) {

        Select s = new Select(element);
        s.selectByVisibleText(text);
    }}
}
```

3.2.5 BaseClass

CUCUMBER

```
package org.pojo;

import java.util.List;

import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

import com.libglobal.BaseClass;

public class BookHotelPOJOClass extends BaseClass {

    public BookHotelPOJOClass() {
        PageFactory.initElements(driver, this);
    }

    @FindBy(id = "first_name")
    private List<WebElement> fname;
    @FindBy(id = "last_name")
    private List<WebElement> lname;
    @FindBy(id = "address")
    private List<WebElement> Address;
    @FindBy(id = "cc_num")
    private List<WebElement> ccnum;
    @FindBy(id = "cc_type")
    private List<WebElement> cardtype;
    @FindBy(id = "cc_exp_month")
    private List<WebElement> month;
    @FindBy(id = "cc_exp_year")
    private List<WebElement> year;
    @FindBy(id = "cc_cvv")
    private List<WebElement> cvv;
    @FindBy(id = "book_now")
    private List<WebElement> book;

    public List<WebElement> getFname() {
        return fname;
    }

    public List<WebElement> getLname() {
        return lname;
    }

    public List<WebElement> getAddress() {
        return Address;
    }

    public List<WebElement> getCcnum() {
        return ccnum;
    }

    public List<WebElement> getCardtype() {
        return cardtype;
    }

    public List<WebElement> getMonth() {
        return month;
    }
}
```

CUCUMBER

```
public List<WebElement> getYear() {  
    return year;  
}  
  
public List<WebElement> getCvv() {  
    return cvv;  
}  
  
public List<WebElement> getBook() {  
    return book;  
}  
  
}
```

3.2.6 BookHotelPOJOClass

```
package org.pojo;  
  
import java.util.List;  
  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.support.FindBy;  
import org.openqa.selenium.support.PageFactory;  
  
import com.libglobal.BaseClass;  
  
public class ConfirmationPOJOClass extends BaseClass {  
  
    public ConfirmationPOJOClass() {  
        PageFactory.initElements(driver, this);  
    }  
  
    @FindBy(id = "search_hotel")  
    private List<WebElement> searchhotel;  
  
    public List<WebElement> getSearchhotel() {  
        return searchhotel;  
    }  
  
}
```

3.2.7 ConfirmationPOJOClass

```
package org.pojo;  
  
import java.util.List;  
  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.support.FindBy;  
import org.openqa.selenium.support.PageFactory;  
  
import com.libglobal.BaseClass;  
  
public class HotelConfirmPOJOClass extends BaseClass {  
  
    public HotelConfirmPOJOClass() {  
        PageFactory.initElements(driver, this);  
    }  
  
}
```


CUCUMBER

```
@FindBy(id = "radiobutton_0")
private List<WebElement> radio;
@FindBy(id = "continue")
private List<WebElement> Continue;

public List<WebElement> getRadio() {
    return radio;
}

public List<WebElement> getContinue() {
    return Continue;
}
}
```

3.2.8 HotelConfirmPOJOClass

```
package org.pojo;

import java.util.List;

import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

import com.libglobal.BaseClass;

public class LoginPOJOClass extends BaseClass {

    public LoginPOJOClass() {
        PageFactory.initElements(driver, this);
    }

    @FindBy(id = "username")
    private List<WebElement> user;
    @FindBy(id = "password")
    private List<WebElement> pass;
    @FindBy(id = "login")
    private List<WebElement> log;

    public List<WebElement> getUser() {
        return user;
    }

    public List<WebElement> getPass() {
        return pass;
    }

    public List<WebElement> getLog() {
        return log;
    }
}
```

3.2.8 LoginPOJOClass

CUCUMBER

```
package org.pojo;
import java.util.List;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import com.libglobal.BaseClass;

public class SearchHotelPOJOClass extends BaseClass {

    public SearchHotelPOJOClass() {

        PageFactory.initElements(driver, this);
    }

    @FindBy(id = "location")
    private List<WebElement> location;
    @FindBy(id = "hotels")
    private List<WebElement> hotels;
    @FindBy(id = "room_type")
    private List<WebElement> roomtype;
    @FindBy(id = "room_nos")
    private List<WebElement> rooms;
    @FindBy(id = "adult_room")
    private List<WebElement> adult_room;
    @FindBy(id = "child_room")
    private List<WebElement> child_room;
    @FindBy(id = "Submit")
    private List<WebElement> submit;

    public List<WebElement> getLocation() {
        return location;
    }

    public List<WebElement> getHotels() {
        return hotels;
    }

    public List<WebElement> getRoomtype() {
        return roomtype;
    }

    public List<WebElement> getRooms() {
        return rooms;
    }

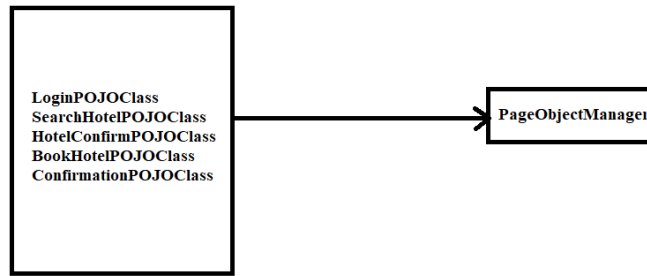
    public List<WebElement> getAdult_room() {
        return adult_room;
    }

    public List<WebElement> getChild_room() {
        return child_room;
    }

    public List<WebElement> getSubmit() {
        return submit;
    }
}
```

3.2.9 SearchHotelPOJOClass

CUCUMBER



PageObjectManager

```
package com.page;

import org.pojo.BookHotelPOJOClass;
import org.pojo.ConfirmationPOJOClass;
import org.pojo.HotelConfirmPOJOClass;
import org.pojo.LoginPOJOClass;
import org.pojo.SearchHotelPOJOClass;

public class PageObjectManager {

    LoginPOJOClass loginPage;
    SearchHotelPOJOClass serchPage;
    HotelConfirmPOJOClass hotelConfirmPage;
    BookHotelPOJOClass bookingPage;
    ConfirmationPOJOClass confirmPage;

    public LoginPOJOClass getLoginPage() {
        return (loginPage == null) ? loginPage = new LoginPOJOClass() : loginPage;
    }

    public SearchHotelPOJOClass getSerchPage() {
        return (serchPage == null) ? serchPage = new SearchHotelPOJOClass() : serchPage;
    }

    public HotelConfirmPOJOClass getHotelConfirmPage() {
        return (hotelConfirmPage == null) ? hotelConfirmPage = new HotelConfirmPOJOClass() :
        hotelConfirmPage;
    }

    public BookHotelPOJOClass getBookingPage() {
        return (bookingPage == null) ? bookingPage = new BookHotelPOJOClass() : bookingPage;
    }
}
```

CUCUMBER

```
public ConfirmationPOJOClass getConfirmPage() {  
    return (confirmPage == null) ? confirmPage = new ConfirmationPOJOClass() :  
confirmPage;  
}  
  
}
```

3.2.10 PageObjectManager

```
Package com.stepdefinition;  
  
import com.libglobal.BaseClass;  
import com.page.PageObjectManager;  
  
import io.cucumber.java.en.*;  
public class StepDefinition extends BaseClass {  
  
    PageObjectManager page = new PageObjectManager();  
  
    @Given("User is on adactin page")  
    public void user_is_on_adactin_page() {  
        launchBrowser();  
        loadUrl("https://adactin.com/HotelApp/");  
        maximizeWindow();  
    }  
  
    @When("User enters {string}and{string}")  
    public void user_enters_and(String userName, String password) {  
  
        type(page.getLoginPage().getUser().get(0), userName);  
        type(page.getLoginPage().getPass().get(0), password);  
  
    }  
  
    @When("User clicks login button")  
    public void user_clicks_login_button() {  
  
        btnClick(page.getLoginPage().getLog().get(0));  
  
    }  
  
    @Then("User verify success message")  
    public void user_verify_success_message() {  
        System.out.println("Successfully login");  
    }  
  
    @When("User select {string}and{string}and{string}and{string}and{string}and{string}")  
    public void user_select_and_and_and_and_and(String location, String hotel, String roomType,  
String numOfRooms,String adultsPerRoom, String childperRoom) {  
  
        selectByVisibleText(page.getSerchPage().getLocation().get(0), location);  
        selectByVisibleText(page.getSerchPage().getHotels().get(0), hotel);  
        selectByVisibleText(page.getSerchPage().getRoomtype().get(0), roomType);  
        selectByVisibleText(page.getSerchPage().getRooms().get(0), numOfRooms);  
    }  
}
```

CUCUMBER

```
        selectByVisibleText(page.getSerchPage().getAdult_room().get(0), adultsPerRoom);
        selectByVisibleText(page.getSerchPage().getChild_room().get(0), childperRoom);
    }

    @When("User click search button")
    public void user_click_search_button() {
        btnClick(page.getSerchPage().getSubmit().get(0));
    }

    @When("User select the Hotel")
    public void user_select_the_Hotel() {

        btnClick(page.getHotelConfirmPage().getRadio().get(0));
    }

    @When("User click the continue button")
    public void user_click_the_continue_button() {
        btnClick(page.getHotelConfirmPage().getContinue().get(0));
    }

    @When("User enter
{string}and{string}and{string}and{string}and{string}and{string}and{string}and{string}")
    public void user_enter_and_and_and_and_and_and_and(String firstNmae, String lastName, String
billingAddress,String creditCardNo, String creditCardType, String expiryMonth, String expiryYear,
String cvvNumber) {

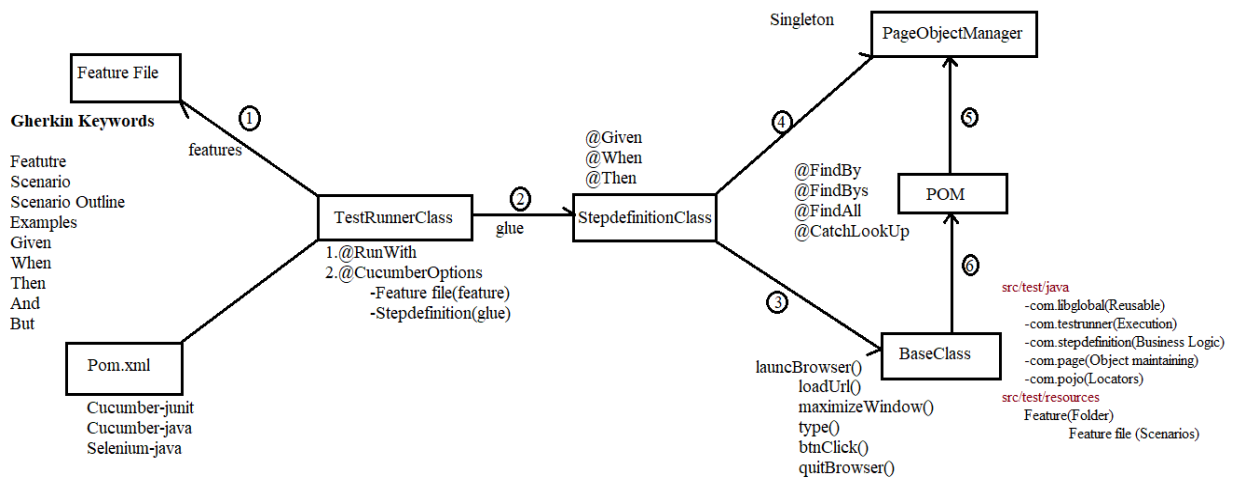
        type(page.getBookingPage().getFname().get(0), firstNmae);
        type(page.getBookingPage().getLname().get(0), lastName);
        type(page.getBookingPage().getAddress().get(0), billingAddress);
        type(page.getBookingPage().getCcnum().get(0), creditCardNo);
        selectByVisibleText(page.getBookingPage().getCardtype().get(0), creditCardType);
        selectByVisibleText(page.getBookingPage().getMonth().get(0), expiryMonth);
        selectByVisibleText(page.getBookingPage().getYear().get(0), expiryYear);
        type(page.getBookingPage().getCvv().get(0), cvvNumber);
    }

    @When("User click BookNow button")
    public void user_click_BookNow_button() {
        btnClick(page.getBookingPage().getBook().get(0));
    }

    @When("User click search hotel")
    public void user_click_search_hotel() throws InterruptedException {
        Thread.sleep(5000);
        btnClick(page.getConfirmPage().getSearchhotel().get(0));
        quitBrowser();
    }
}
```

3.2.11 StepDefinitionClass

CUCUMBER



3.2.12 Cucumber Architecture

TO PASS THE PARAMETER USING SQL IN CUCUMBER

1. Create maven project and add ojdbc jar.
2. Pass the JDBC connection in base class.

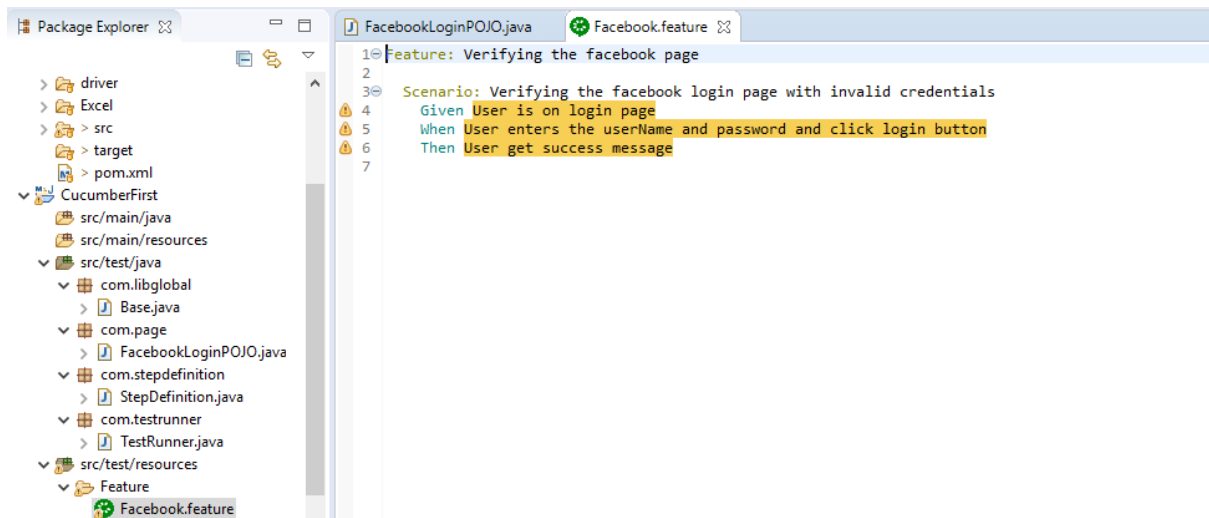
The screenshot shows an IDE with the following components:

- Package Explorer:** Shows the project structure with a package named `com.testrunner`.
- Console:** Displays the output of the test run, showing "Finished after 17.284 seconds" and "Runs: 1/1 Errors: 0 Failures: 0".
- Code Editor:** Shows the `TestRunner.java` file with the following code:

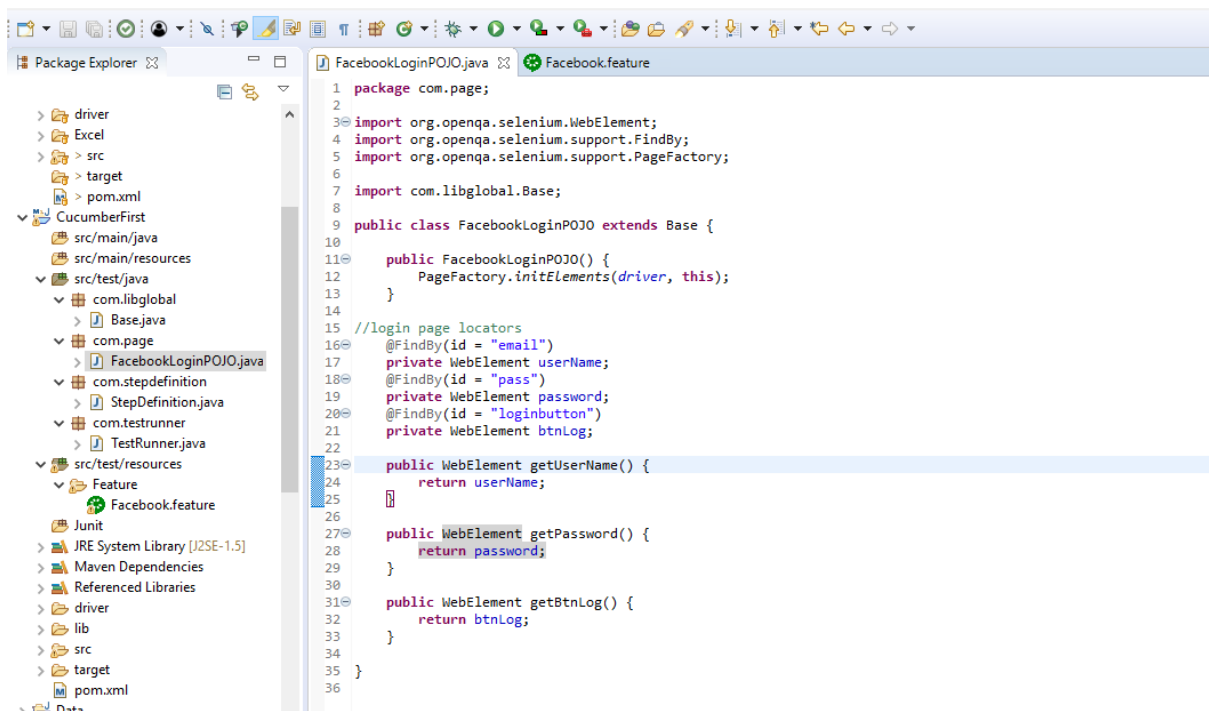
```
1 package com.testrunner;
2
3 import org.junit.runner.RunWith;
4
5 import io.cucumber.junit.Cucumber;
6 import io.cucumber.junit.CucumberOptions;
7
8 @RunWith(Cucumber.class)
9 @CucumberOptions(features = { "src/test/resources/Feature" }, glue = { "com.stepdefinition" })
10 public class TestRunner {
11
12 }
13 }
```

TestRunnerClass

CUCUMBER



Feature File



PojoClass

```
package com.libglobal;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

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

public class Base {
    public static WebDriver driver;
```

CUCUMBER

```
public static void launchBrowser() {
    System.setProperty("webdriver.chrome.driver",
        "C:\\\\Users\\k_sur\\eclipseworkspace\\CucumberFirst\\driver\\chromedriver.exe");
    driver = new ChromeDriver();
}

public static void maximizeWindow() {
    driver.manage().window().maximize();
}

public static void loadUrl(String url) {
    driver.get(url);
}

public static void type(WebElement element, String name) {
    element.sendKeys(name);
}

public static void btnClick(WebElement e) {
    e.click();
}

public static void quitBrowser() {
    driver.quit();
}

public static String getDataFromSql(String query) throws Throwable {
    Connection con = null;
    String name = null;
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "hr", "admin");

        PreparedStatement ps = con.prepareStatement(query);
        ResultSet rs = ps.executeQuery();
        while (rs.next()) {
            name = rs.getString("first_name");
        }

    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        try {
            con.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    return name;
}
}
```

BaseClass

CUCUMBER

```
package com.stepdefinition;

import com.libglobal.Base;
import com.page.FacebookLoginPOJO;

import io.cucumber.java.en.*;

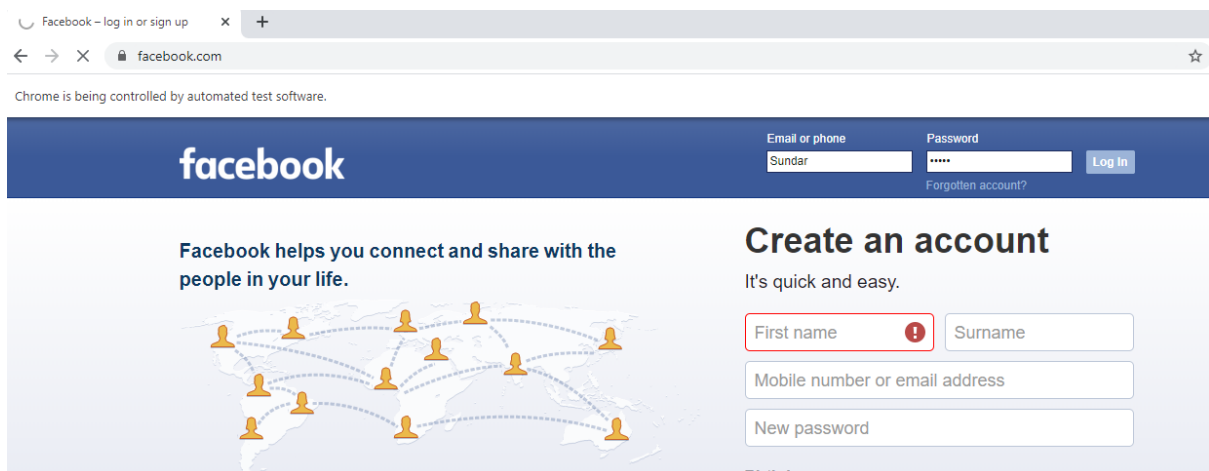
public class StepDefinition extends Base {

    @Given("User is on login page")
    public void user_is_on_login_page() {
        LaunchBrowser();// launch the browser
        maximizeWindow();// maximize the window
        LoadUrl("https://www.facebook.com/");
    }

    @When("User enters the userName and password and click login button")
    public void user_enters_the_userName_and_password_and_click_login_button() throws Throwable {
        FacebookLoginPOJO f = new FacebookLoginPOJO();
        // get data from sql
        type(f.getUserName(), getDataFromSql("select first_name from employees where first_name='Sundar'"));
        type(f.getPassword(), getDataFromSql("select first_name from employees where first_name='Ellen'"));
        btnClick(f.getBtnLog());// clicks the login button
        quitBrowser();// quits the browser
    }

    @Then("User get success message")
    public void user_get_success_message() {
        System.out.println("done...");
    }
}
```

StepDefinitionClass

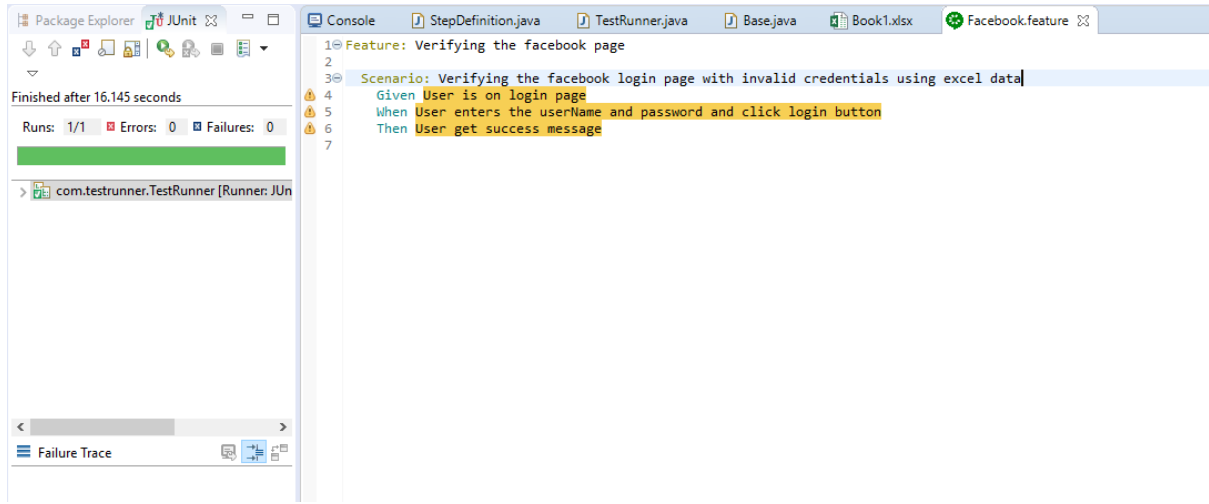


Output

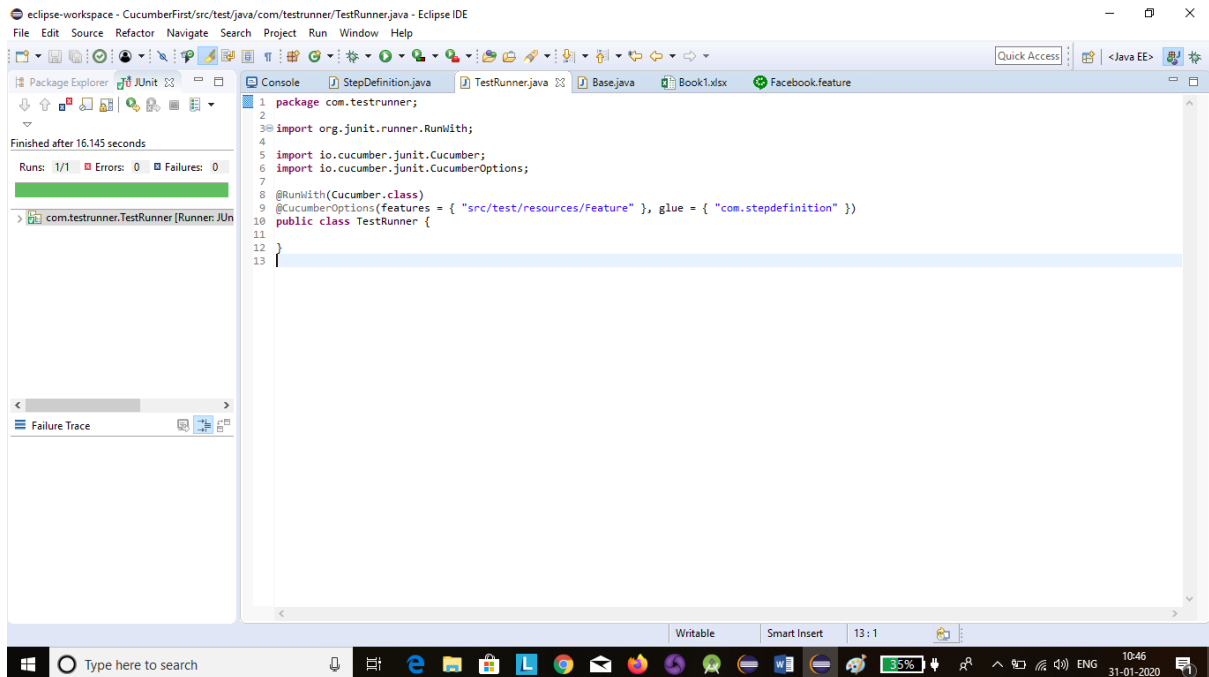
CUCUMBER

TO PASSING THE PARAMETER USING EXCEL IN CUCUMBER

1. Create Maven project add poi-ooxml,commons-io dependencies.
2. Create Folder and copy paste the Excel.



Feature file



TestRunnerClass

CUCUMBER

```
package com.libglobal;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;

import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.DateUtil;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;

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

public class Base {
    public static WebDriver driver;

    public static void launchBrowser() {
        System.setProperty("webdriver.chrome.driver",
            "C:\\\\Users\\k_sur\\eclipse-
workspace\\CucumberFirst\\driver\\chromedriver.exe");
        driver = new ChromeDriver();
    }

    public static void maximizeWindow() {
        driver.manage().window().maximize();
    }

    public static void loadUrl(String url) {
        driver.get(url);
    }

    public static void type(WebElement element, String name) {
        element.sendKeys(name);
    }

    public static void btnClick(WebElement e) {
        e.click();
    }

    public static void quitBrowser() {
        driver.quit();
    }

    @SuppressWarnings({ "deprecation", "resource" })
    public static String getDataFromExcel(int row, int cell) throws IOException {
        String value = null;
        File loc = new
File("C:\\\\Users\\k_sur\\eclipseworkspace\\CucumberFirst\\Excel\\Book1.xlsx");
```

CUCUMBER

```
FileInputStream stream = new FileInputStream(loc);
Workbook w = new XSSFWorkbook(stream);
Sheet s = w.getSheet("Greens");
Row r = s.getRow(row);
Cell c = r.getCell(cell);

int type = c.getCellType();

if (type == 1) {

    value = c.getStringCellValue();

} else if (type == 0) {
    if (DateUtil.isCellDateFormatted(c)) {
        Date dateCellValue = c.getDateCellValue();
        SimpleDateFormat sim = new SimpleDateFormat("dd-mm-yyyy");
        value = sim.format(dateCellValue);

    } else {

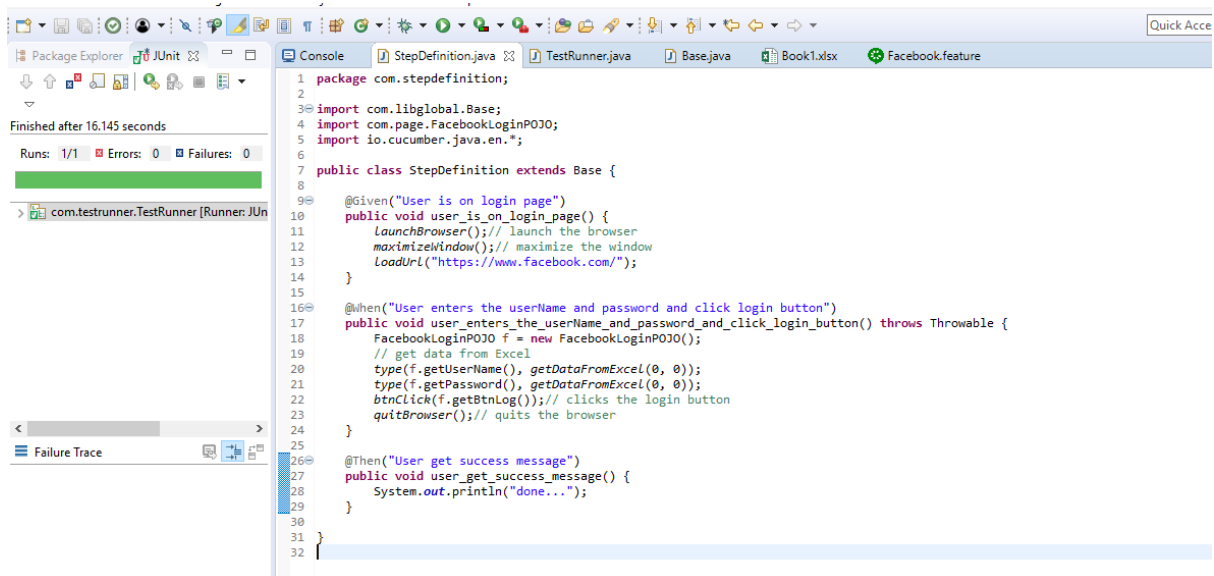
        double numericCellValue = c.getNumericCellValue();
        long l = (long) numericCellValue;
        value = String.valueOf(l);

    }

}

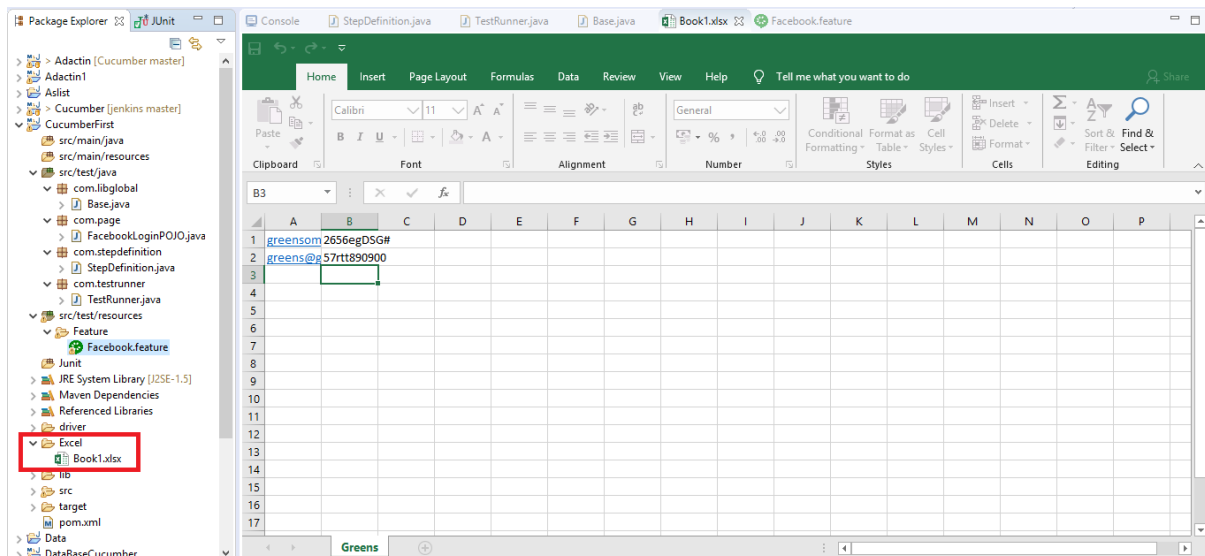
return value;
}
```

BaseClass

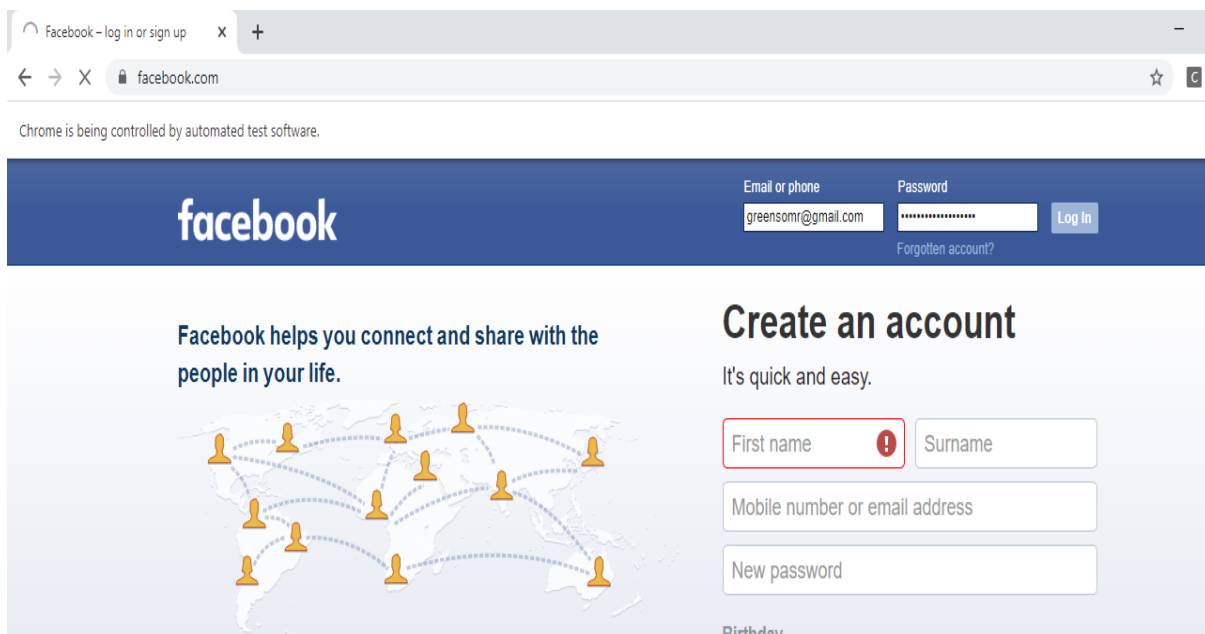


StepDefinitionClass

CUCUMBER



Excel File



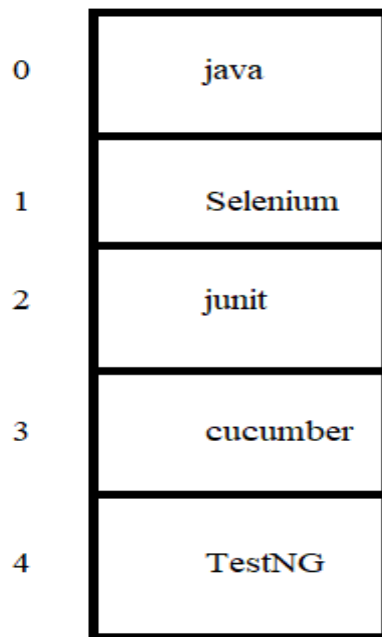
Output

DATA TABLE IN CUCUMBER

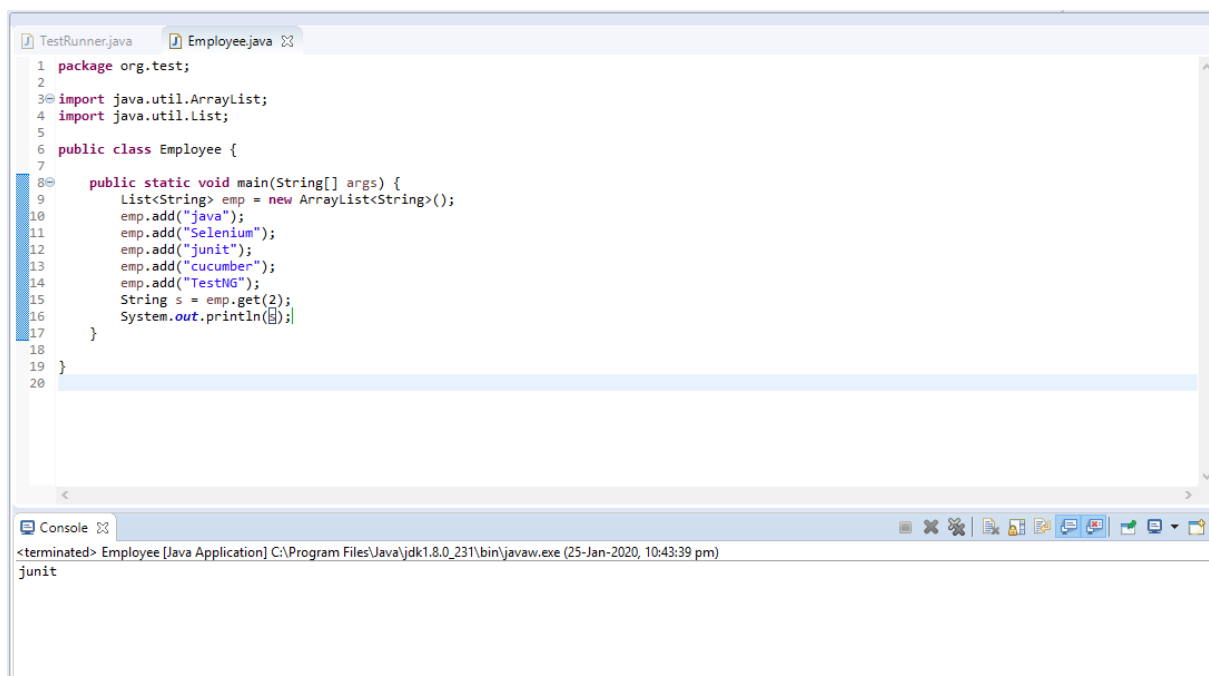
1. asList(1D-Without Header).
2. asLists(2D-Without Header).
3. asMap(1D-With Header).
4. asMaps(2D-With Header).

CUCUMBER

1.1D-WITHOUT HEADER(asList)



1D –List Diagram



The screenshot shows an IDE with two tabs: TestRunner.java and Employee.java. The Employee.java tab is active, displaying the following code:

```
1 package org.test;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Employee {
7
8     public static void main(String[] args) {
9         List<String> emp = new ArrayList<String>();
10        emp.add("java");
11        emp.add("Selenium");
12        emp.add("junit");
13        emp.add("cucumber");
14        emp.add("TestNG");
15        String s = emp.get(2);
16        System.out.println(s);
17    }
18 }
19
20
```

The console output at the bottom shows the execution of the program:

```
<terminated> Employee [Java Application] C:\Program Files\Java\jdk1.8.0_231\bin\javaw.exe (25-Jan-2020, 10:43:39 pm)
junit
```

1D-Without Header

CUCUMBER

2.2D-WITHOUT HEADER(asLists)

Column Index

|

`List<ArrayList<String>> sam = new LinkedList<ArrayList<String>>();`

|

Row Index

	0	1	2
0	bala	bala@gmail.com	selenium
1	karthi	karthi@gmail.com	python
2	kiruba	kiruba@gmail.com	sql

2D-List Diagram

```
Sample.java
1 package org.test;
2
3 import java.util.ArrayList;
4 import java.util.LinkedList;
5 import java.util.List;
6
7 public class Sample {
8
9     public static void main(String[] args) {
10         // Outer list
11         List<ArrayList<String>> sam = new LinkedList<ArrayList<String>>();
12         // Inner list
13         ArrayList<String> s1 = new ArrayList<String>();
14         s1.add("bala");
15         s1.add("bala@gmail.com");
16         s1.add("selenium");
17         ArrayList<String> s2 = new ArrayList<String>();
18         s2.add("karthi");
19         s2.add("karthi@gmail.com");
20         s2.add("python");
21         ArrayList<String> s3 = new ArrayList<String>();
22         s3.add("kiruba");
23         s3.add("kiruba@gmail.com");
24         s3.add("sql");
25         sam.add(s1);
26         sam.add(s2);
27         sam.add(s3);
28         // get the values from outer list
29         ArrayList<String> list = sam.get(1);
30         // get the values from inner list
31         String string = list.get(1);
32         System.out.println(string);
33     }
34 }
35
36
```

Console

<terminated> Sample (3)
karthi@gmail.com

2D-Without Header

CUCUMBER

3.1D-WITH HEADER(asMap)

Data (or) Column Data (or) Value

↓

```
Map<String, String> mp = new LinkedHashMap<String, String>();
```

↓

Header Name (or) Column Name (or) Key

name	bala
gmail	bala@gmail.com
phoneNo	9659354251

1D-Map Diagram

The screenshot shows an IDE with two tabs: 'Test.java' and 'Employee1.java'. The 'Test.java' tab is active, displaying the following code:

```
1 package org.test;
2
3 import java.util.LinkedHashMap;
4 import java.util.Map;
5
6 public class Test {
7
8     public static void main(String[] args) {
9
10         Map<String, String> mp = new LinkedHashMap<String, String>();
11         mp.put("name", "bala");
12         mp.put("gmail", "bala@gmail.com");
13         mp.put("phoneNo", "9659354251");
14         String s = mp.get("gmail");
15         System.out.println(s);
16     }
17 }
18
19 }
```

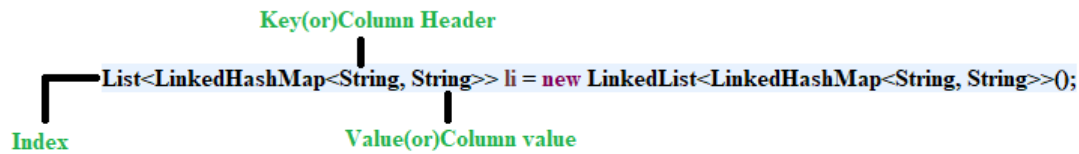
The console output on the right shows the result of the program execution:

```
<terminated> Test (2) [Java]
bala@gmail.com
```

1D-With Header

CUCUMBER

3.2D-WITH HEADER(asMaps)



	name	gmail	phoneNo
0	nitish	nitish2198@gmail.com	7896541230
1	merin	prbmerin@gmail.com	9321654870

2D-Map Diagram

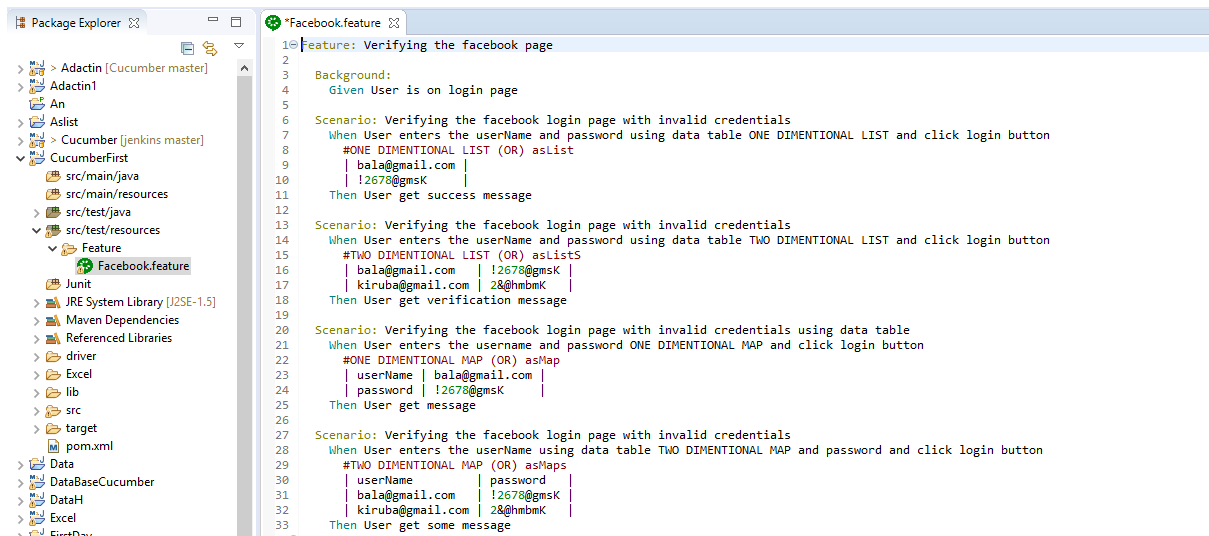
```
Testjava  AsMaps.java
1 package org.test;
2
3 import java.util.LinkedHashMap;
4
5
6
7 public class AsMaps {
8
9     public static void main(String[] args) {
10         // outer list declaration
11         List<LinkedHashMap<String, String>> li = new LinkedList<LinkedHashMap<String, String>>();
12         // inner map declaration
13         LinkedHashMap<String, String> mp1 = new LinkedHashMap<String, String>();
14         mp1.put("name", "nitish");
15         mp1.put("gmail", "nitish2198@gmail.com");
16         mp1.put("phoneNo", "7896541230");
17         LinkedHashMap<String, String> mp2 = new LinkedHashMap<String, String>();
18         mp2.put("name", "merin");
19         mp2.put("gmail", "prbmerin@gmail.com");
20         mp2.put("phoneNo", "9321654870");
21         li.add(mp1);
22         li.add(mp2);
23         LinkedHashMap<String, String> asMaps = li.get(1);
24         String s = asMaps.get("gmail");
25         System.out.println(s);
26     }
27 }
28
29
```

Console Output:

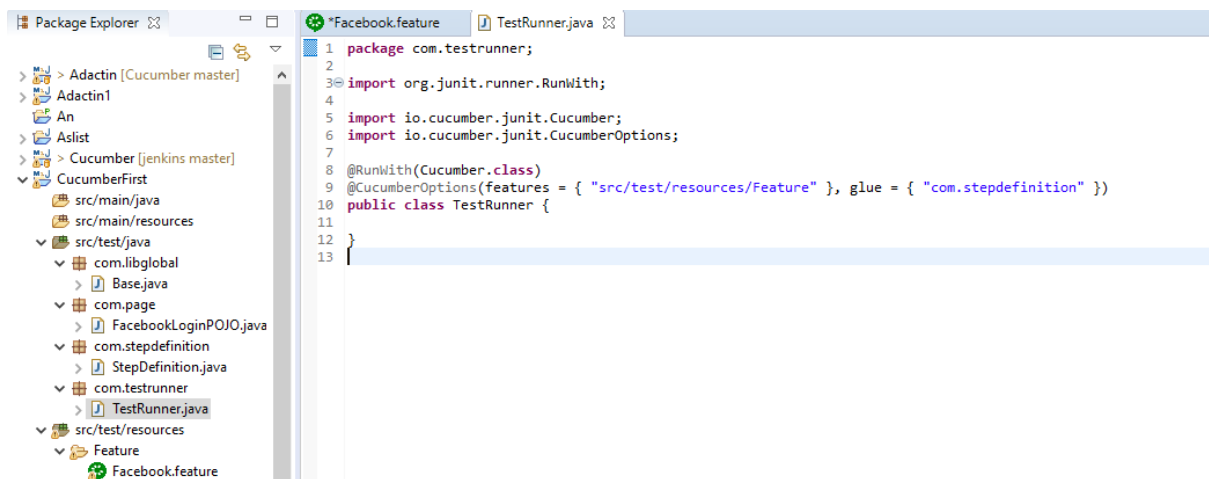
```
<terminated> AsMaps [Java
prbmerin@gmail.com
```

2D-With Header

CUCUMBER



Feature File



TestRunnerClass

```
package com.libglobal;

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

public class Base {
    public static WebDriver driver;

    public static void launchBrowser() {
        System.setProperty("webdriver.chrome.driver",
            "C:\\\\Users\\k_sur\\eclipse-
workspace\\CucumberFirst\\driver\\chromedriver.exe");
        driver = new ChromeDriver();
    }

    public static void maximizeWindow() {
        driver.manage().window().maximize();
    }
}
```

CUCUMBER

```
public static void loadUrl(String url) {
    driver.get(url);
}

public static void type(WebElement element, String name) {
    element.sendKeys(name);
}

public static void btnClick(WebElement e) {
    e.click();
}

public static void quitBrowser() {
    driver.quit();
}
}
```

BaseClass

```
package com.page;

import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

import com.libglobal.Base;

public class FacebookLoginPOJO extends Base {
    public FacebookLoginPOJO() {
        PageFactory.initElements(driver, this);
    }

    //login page locators
    @FindBy(id = "email")
    private WebElement userName;
    @FindBy(id = "pass")
    private WebElement password;
    @FindBy(id = "loginbutton")
    private WebElement btnLog;

    public WebElement getUsername() {
        return userName;
    }

    public WebElement getPassword() {
        return password;
    }

    public WebElement getBtnLog() {
        return btnLog;
    }
}
```

Login POJO

CUCUMBER

```
package com.stepdefinition;

import java.util.List;
import java.util.Map;

import com.libglobal.Base;
import com.page.FacebookLoginPOJO;

import io.cucumber.java.en.*;

public class StepDefinition extends Base {
    FacebookLoginPOJO f;

    @Given("User is on login page")
    public void user_is_on_login_page() {
        launchBrowser();
        loadUrl("https://www.facebook.com/");
        maximizeWindow();
    }

    @When("User enters the userName and password using data table ONE DIMENTIONAL LIST and click login button")
    public void user_enters_the_userName_and_password_using_data_table_ONE_DIMENTIONAL_LIST_and_click_login_button(
        io.cucumber.datatable.DataTable dataFromList) {
        f = new FacebookLoginPOJO();
        List<String> list = dataFromList.asList();
        type(f.getUserName(), list.get(0)); // 1d without header
        type(f.getPassword(), list.get(1)); // 1d without header
        btnClick(f.getBtnLog());
    }

    @Then("User get success message")
    public void user_get_success_message() {
        quitBrowser();
        System.out.println("done...");
    }

    @When("User enters the userName and password using data table TWO DIMENTIONAL LIST and click login button")
    public void user_enters_the_userName_and_password_using_data_table_TWO_DIMENTIONAL_LIST_and_click_login_button(
        io.cucumber.datatable.DataTable dataFromLists) {
        f = new FacebookLoginPOJO();
        List<List<String>> lists = dataFromLists.asLists();
        type(f.getUserName(), lists.get(0).get(0)); // 2d without header
        type(f.getPassword(), lists.get(0).get(1)); // 2d without header
        btnClick(f.getBtnLog());
    }

    @Then("User get verification message")
    public void user_get_verification_message() {
        quitBrowser();
        System.out.println("done...");
    }
}
```

CUCUMBER

```
@When("User enters the username and password ONE DIMENTIONAL MAP and click login button")
public void user_enters_the_username_and_password_ONE_DIMENTIONAL_MAP_and_click_login_button(
    io.cucumber.datatable.DataTable dataFromMap) {
    f = new FacebookLoginPOJO();
    Map<String, String> map = dataFromMap.asMap(String.class, String.class);
    type(f.getUserName(), map.get("userName")); // 1d with header
    type(f.getPassword(), map.get("password")); // 1d with header
    btnClick(f.getBtnLog());
}

@Then("User get message")
public void user_get_message() {
    quitBrowser();
    System.out.println("done...");
}

@When("User enters the userName using data table TWO DIMENTIONAL MAP and password and click login button")
public void user_enters_the_userName_using_data_table_TWO_DIMENTIONAL_MAP_and_password_and_click_login_button(
    io.cucumber.datatable.DataTable dataFromMaps) {
    f = new FacebookLoginPOJO();
    List<Map<String, String>> maps = dataFromMaps.asMaps();

    type(f.getUserName(), maps.get(1).get("userName")); // 2d with header
    type(f.getPassword(), maps.get(0).get("password")); // 2d with header
    btnClick(f.getBtnLog());
}

@Then("User get some message")
public void user_get_some_message() {
    quitBrowser();
    System.out.println("done...");
}
}
```

Stepdefinition Class

HOOKS IN CUCUMBER

- Cucumber supports **hooks** which are blocks of code that run **before** or **after** each scenario. **Cucumber Hooks** allows us to better manage the code workflow and helps as to reduce the code redundancy.
- In hooks we will be having two annotations **@Before** and **@After**.
- In Hooks we can prioritize the execution order.
- In Hooks we can use tags (tagged hooks).

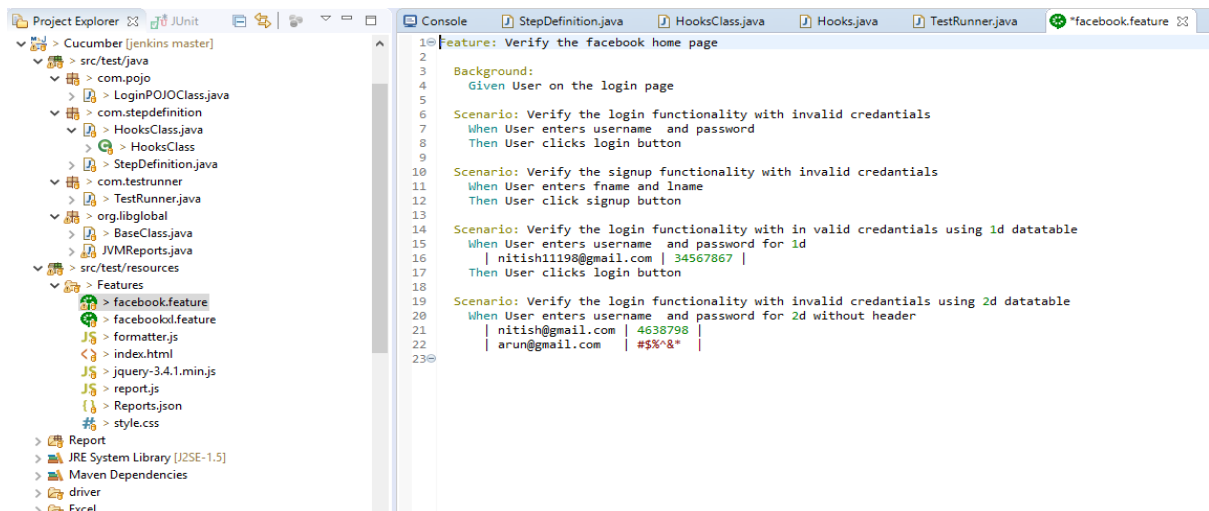
CUCUMBER

@Before

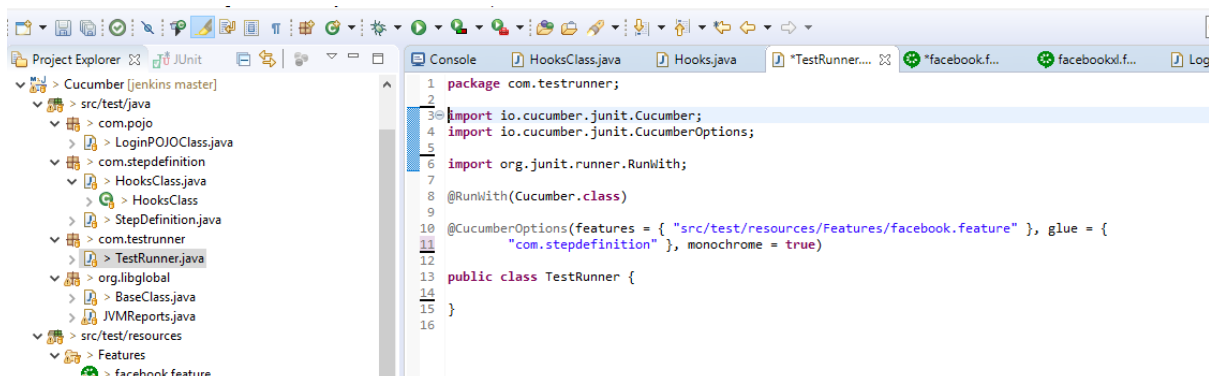
- Which is going to execute before each scenario like
 1. Launch browser.
 2. Load url.
 3. Delete cookies.
 4. Maximize window.
 5. Waits.

@After

- Which is going to execute after each scenario like
 1. Close browser.
 2. Quit browser.
 3. Screenshot.
 4. Closing the database connection.

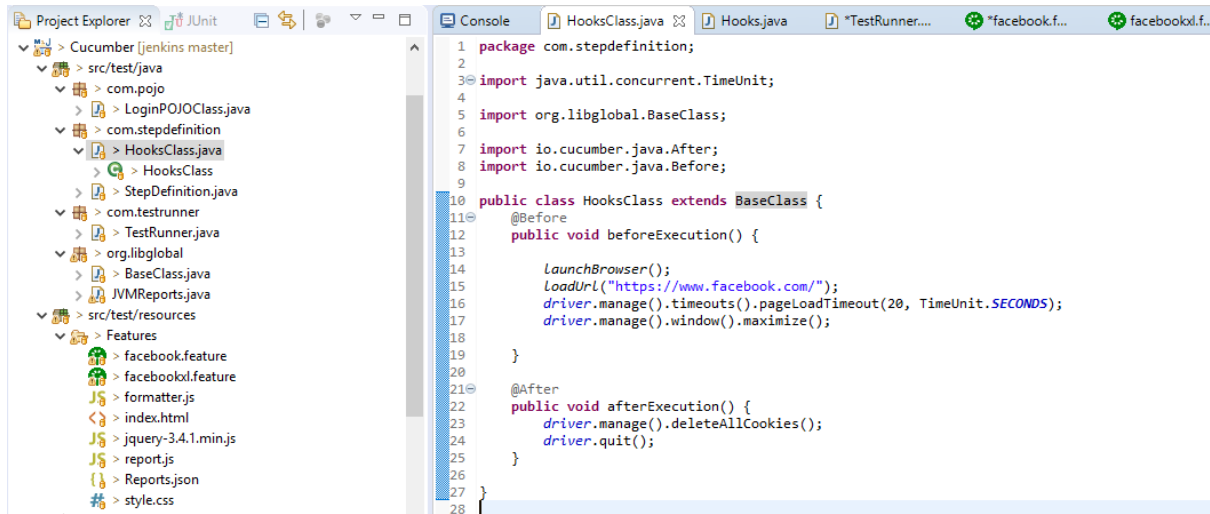


Feature File One



TestRunner Class

CUCUMBER



HooksClass

```
package org.libglobal;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;

import org.apache.commons.io.FileUtils;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.DateUtil;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;

public class BaseClass {
    public static WebDriver driver;

    public static void launchBrowser() {
        System.setProperty("webdriver.chrome.driver",
            "C:\\\\Users\\k_sur\\Desktop\\Nitish\\cucumber
jenkins\\jenkins\\driver\\chromedriver.exe");
        driver = new ChromeDriver();
    }

    public static void maximizeWindow() {
        driver.manage().window().maximize();
    }
}
```

CUCUMBER

```
}

public static void loadUrl(String url) {
    driver.get(url);
}

public static void type(WebElement e, String value) {
    e.sendKeys(value);
}

public static void btnClick(WebElement e) {
    e.click();
}

public static String getTittle() {
    String title = driver.getTitle();
    return title;
}

public static String getUrl() {
    String Url = driver.getCurrentUrl();
    return Url;
}

public static void quitBrowser() {
    driver.quit();
}

public static void selectByVisibleText(WebElement element, String text) {
    new Select(element).selectByVisibleText(text);
}

public static String getTextAttribute(WebElement element) {
    String attribute = element.getAttribute("value");
    return attribute;
}

@SuppressWarnings({ "deprecation", "resource" })
public static String getDataFromExcel(int row, int cell) throws IOException {
    String value = null;
    File loc = new File("C:\\Users\\k_sur\\eclipse-
workspace\\Cucumber\\Excel\\integration.xlsx");
    FileInputStream stream = new FileInputStream(loc);
    Workbook w = new XSSFWorkbook(stream);
    Sheet s = w.getSheet("Sheet1");
    Row r = s.getRow(row);
    Cell c = r.getCell(cell);
```


CUCUMBER

```
int type = c.getCellType();

if (type == 1) {

    value = c.getStringCellValue();

} else if (type == 0) {
    if (DateUtil.isCellDateFormatted(c)) {
        Date dateCellValue = c.getDateCellValue();
        SimpleDateFormat sim = new SimpleDateFormat("dd-mm-yyyy");
        value = sim.format(dateCellValue);

    } else {

        double numericCellValue = c.getNumericCellValue();
        long l = (long) numericCellValue;
        value = String.valueOf(l);

    }

}

return value;

}
```

BaseClass

```
package com.pojo;

import org.libglobal.BaseClass;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

public class LoginPOJOClass extends BaseClass {

    public LoginPOJOClass() {
        PageFactory.initElements(driver, this);
    }

    @FindBy(id = "email")
    private WebElement username;

    @FindBy(id = "pass")
    private WebElement password;

    @FindBy(id = "loginbutton")
    private WebElement login;

    @FindBy(id = "u_0_m")
    private WebElement fname;

    @FindBy(id = "u_0_o")
    private WebElement lname;

    @FindBy(id = "u_0_13")
    private WebElement signup;

}
```

CUCUMBER

```
public WebElement getUsername() {  
    return username;  
}  
  
public WebElement getPassword() {  
    return password;  
}  
  
public WebElement getLogin() {  
    return login;  
}  
  
public WebElement getFname() {  
    return fname;  
}  
  
public WebElement getLname() {  
    return lname;  
}  
  
public WebElement getSignup() {  
    return signup;  
}  
}
```

LoginPojoClass

```
package com.stepdefinition;  
  
import java.io.IOException;  
import java.util.List;  
import org.libglobal.BaseClass;  
import com.pojo.LoginPOJOClass;  
  
import io.cucumber.java.en.*;  
  
public class StepDefinition extends BaseClass {  
  
    LoginPOJOClass l;  
  
    @Given("User on the login page")  
    public void user_on_the_login_page() {  
  
    }  
  
    @When("User enters username and password")  
    public void user_enters_username_and_password() {  
        l = new LoginPOJOClass();  
  
        type(l.getUsername(), "nitish11198@gmail.com");  
        type(l.getPassword(), "345678");  
  
    }  
  
    @Then("User clicks login button")  
  
    public void user_clicks_login_button() {
```

CUCUMBER

```
        l = new LoginPOJOClass();

        btnClick(l.getLogin());
    }

    @When("User enters fname and lname")
    public void user_enters_fname_and_lname() {
        l = new LoginPOJOClass();

        type(l.getFname(), "nitish");
        type(l.getLname(), "kumar");
    }

    @Then("User click signup button")
    public void user_click_signup_button() {
        l = new LoginPOJOClass();

        btnClick(l.getSignup());
    }

    @When("User enters username and password for {int}d")
    public void user_enters_username_and_password_for_d(Integer int1,
io.cucumber.datatable.DataTable d) {
        List<String> li = d.asList();
        l = new LoginPOJOClass();

        type(l.getUsername(), li.get(0));
        type(l.getPassword(), li.get(1));
    }

    @When("User enters username and password for {int}d without header")
    public void user_enters_username_and_password_for_d_without_header(Integer int1,
        io.cucumber.datatable.DataTable d) {
        List<List<String>> list = d.asLists();
        l = new LoginPOJOClass();

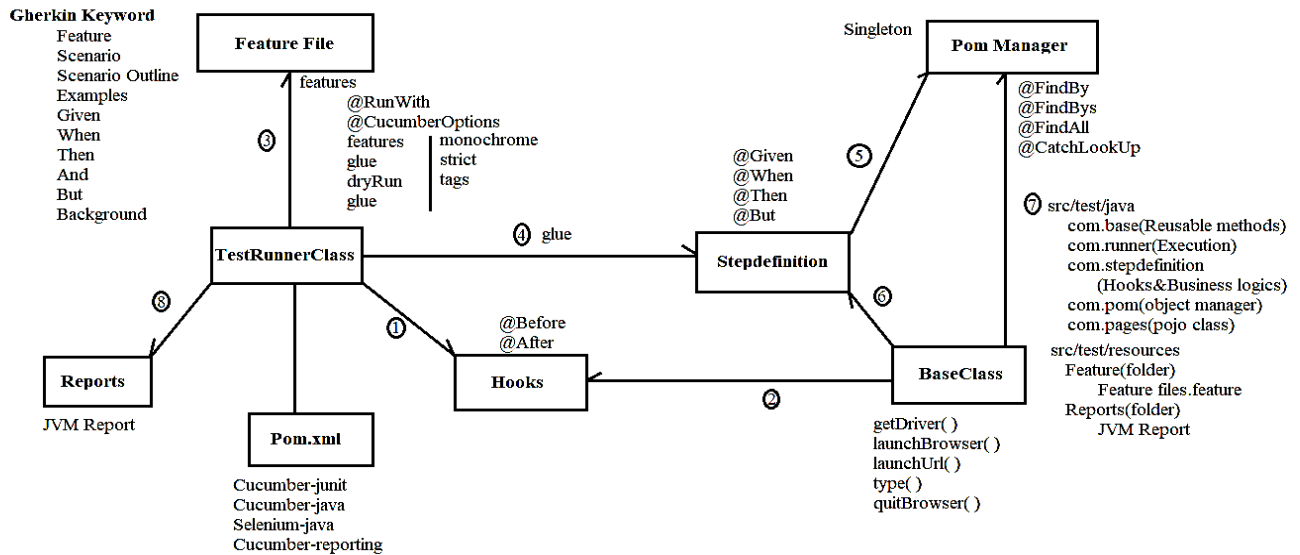
        type(l.getUsername(), list.get(0).get(0));
        type(l.getPassword(), list.get(0).get(1));
    }

    @When("User enters the username and password")
    public void user_enters_the_username_and_password() throws IOException {
        l = new LoginPOJOClass();
        type(l.getUsername(), getDataFromExcel(1, 1));
        type(l.getPassword(), getDataFromExcel(1, 1));
    }

    @Then("User clicks the login button")
    public void user_clicks_the_login_button() {
        btnClick(l.getLogin());
    }
}
```

StepDefinitionClass

CUCUMBER



Cucumber Architecture

CUCUMBER TAGS

- Using tags we can run the bulk of testcases.
- Using tags we can skip the bulk of testcase.

Syntax

➤ tags={"@tagname"}

ignore

➤ tags={"~@tagname"}

groups using OR operator

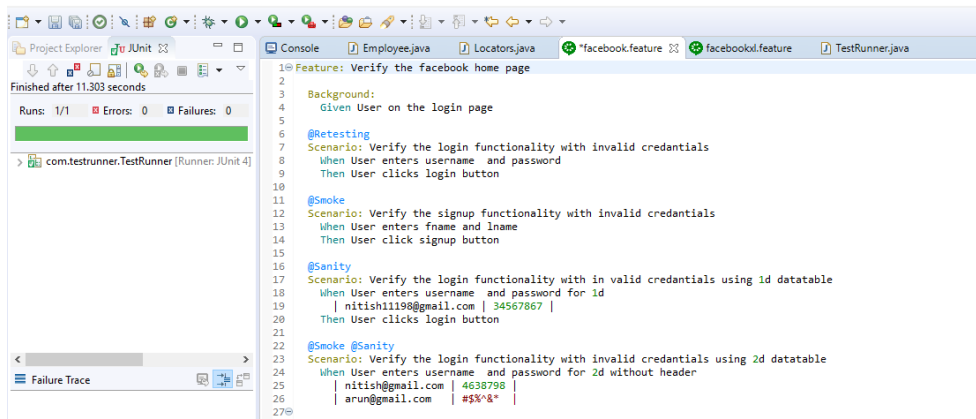
- using OR operator we can run more than one groups.
- Comma (,) operator is used for OR.

tags={"@tagname1 , @tagname2"}

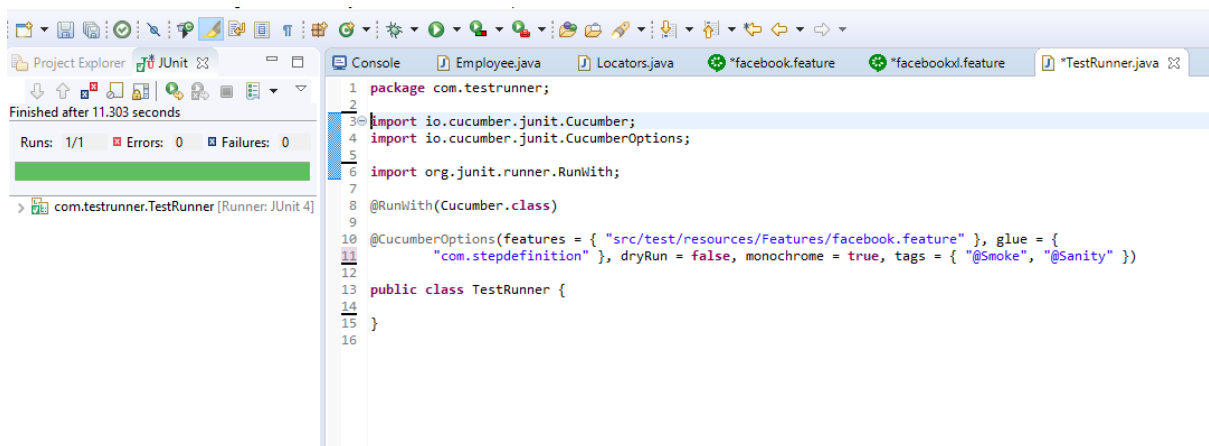
groups using AND operator

➤ Tags={"@tagname1", "@tagname2"}

CUCUMBER

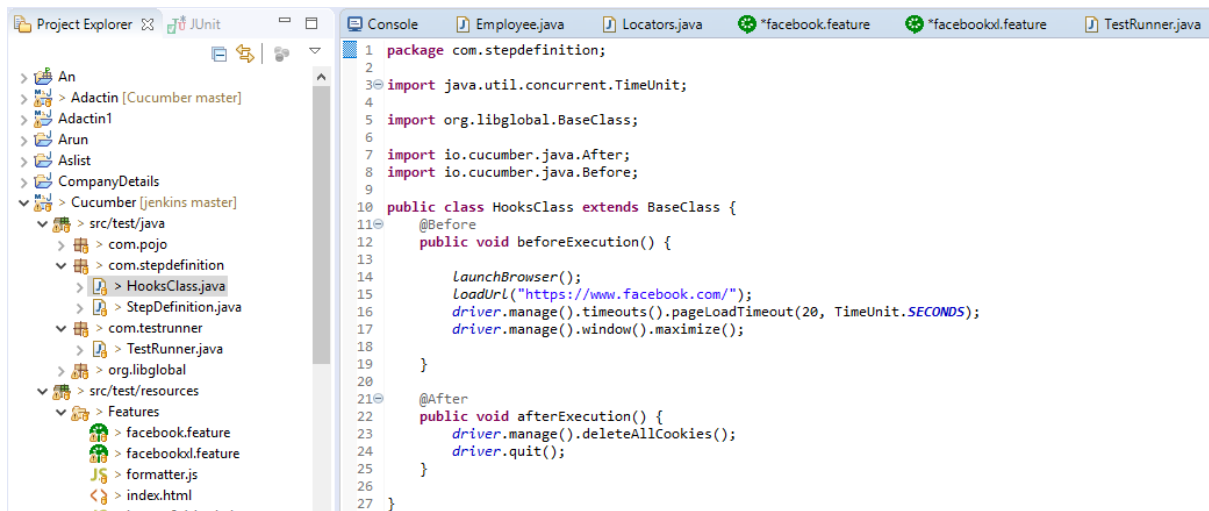


Feature File one



TestRunner Class

CUCUMBER



Hooks Class

```
package org.libglobal;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;

import org.apache.commons.io.FileUtils;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.DateUtil;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;

public class BaseClass {
    public static WebDriver driver;

    public static void launchBrowser() {
        System.setProperty("webdriver.chrome.driver",
            "C:\\Users\\k_sur\\Desktop\\Nitish\\cucumber
jenkins\\jenkins\\driver\\chromedriver.exe");
        driver = new ChromeDriver();
    }

    public static void maximizeWindow() {
        driver.manage().window().maximize();
    }
}
```

CUCUMBER

```
public static void loadUrl(String url) {
    driver.get(url);
}

public static void type(WebElement e, String value) {
    e.sendKeys(value);
}

public static void btnClick(WebElement e) {
    e.click();
}

public static String getTittle() {
    String title = driver.getTitle();
    return title;
}

public static String getUrl() {
    String Url = driver.getCurrentUrl();
    return Url;
}

public static void quitBrowser() {
    driver.quit();
}

public static void selectByVisibleText(WebElement element, String text) {
    new Select(element).selectByVisibleText(text);
}

public static String getTextAttribute(WebElement element) {
    String attribute = element.getAttribute("value");
    return attribute;
}

@SuppressWarnings({ "deprecation", "resource" })
public static String getDataFromExcel(int row, int cell) throws IOException {
    String value = null;
    File loc = new File("C:\\Users\\k_sur\\eclipse-
workspace\\Cucumber\\Excel\\integration.xlsx");
    FileInputStream stream = new FileInputStream(loc);
    Workbook w = new XSSFWorkbook(stream);
    Sheet s = w.getSheet("Sheet1");
    Row r = s.getRow(row);
    Cell c = r.getCell(cell);

    int type = c.getCellType();
```

CUCUMBER

```
        if (type == 1) {

            value = c.getStringCellValue();

        } else if (type == 0) {
            if (DateUtil.isCellDateFormatted(c)) {
                Date dateCellValue = c.getDateCellValue();
                SimpleDateFormat sim = new SimpleDateFormat("dd-mm-yyyy");
                value = sim.format(dateCellValue);

            } else {

                double numericCellValue = c.getNumericCellValue();
                long l = (long) numericCellValue;
                value = String.valueOf(l);

            }

        }

        return value;

    }

}
```

BaseClass

```
package com.pojo;

import org.libglobal.BaseClass;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

public class LoginPOJOClass extends BaseClass {

    public LoginPOJOClass() {
        PageFactory.initElements(driver, this);
    }

    @FindBy(id = "email")
    private WebElement username;

    @FindBy(id = "pass")
    private WebElement password;

    @FindBy(id = "loginbutton")
    private WebElement login;

    @FindBy(id = "u_0_m")
    private WebElement fname;

    @FindBy(id = "u_0_o")
    private WebElement lname;

    @FindBy(id = "u_0_13")
    private WebElement signup;

    public WebElement getUsername() {
        return username;
    }

}
```


CUCUMBER

```
public WebElement getPassword() {  
    return password;  
}  
  
public WebElement getLogin() {  
    return login;  
}  
  
public WebElement getFname() {  
    return fname;  
}  
  
public WebElement getLname() {  
    return lname;  
}  
  
public WebElement getSignup() {  
    return signup;  
}  
}
```

Pojo Class

```
package com.stepdefinition;  
  
import java.io.IOException;  
import java.util.List;  
import org.libglobal.BaseClass;  
import com.pojo.LoginPOJOClass;  
  
import io.cucumber.java.en.*;  
  
public class StepDefinition extends BaseClass {  
  
    LoginPOJOClass l;  
  
    @Given("User on the login page")  
    public void user_on_the_login_page() {  
  
    }  
  
    @When("User enters username and password")  
    public void user_enters_username_and_password() {  
        l = new LoginPOJOClass();  
  
        type(l.getUsername(), "nitish11198@gmail.com");  
        type(l.getPassword(), "345678");  
  
    }  
  
    @Then("User clicks login button")  
  
    public void user_clicks_login_button() {  
        l = new LoginPOJOClass();  
  
        btnClick(l.getLogin());  
    }  
}
```

CUCUMBER

```
}

@When("User enters fname and lname")
public void user_enters_fname_and_lname() {
    l = new LoginPOJOClass();

    type(l.getFname(), "nitish");
    type(l.getLname(), "kumar");

}

@Then("User click signup button")
public void user_click_signup_button() {
    l = new LoginPOJOClass();

    btnClick(l.getSignup());
}

@When("User enters username and password for {int}d")
public void user_enters_username_and_password_for_d(Integer int1,
io.cucumber.datatable.DataTable d) {
    List<String> li = d.asList();
    l = new LoginPOJOClass();

    type(l.getUsername(), li.get(0));
    type(l.getPassword(), li.get(1));
}

@When("User enters username and password for {int}d without header")
public void user_enters_username_and_password_for_d_without_header(Integer int1,
io.cucumber.datatable.DataTable d) {
    List<List<String>> list = d.asLists();
    l = new LoginPOJOClass();

    type(l.getUsername(), list.get(0).get(0));
    type(l.getPassword(), list.get(0).get(1));
}

@When("User enters the username and password")
public void user_enters_the_username_and_password() throws IOException {
    l = new LoginPOJOClass();
    type(l.getUsername(), getDataFromExcel(1, 1));
    type(l.getPassword(), getDataFromExcel(1, 1));
}

@Then("User clicks the login button")
public void user_clicks_the_login_button() {
    btnClick(l.getLogin());
}

}
```

Stepdefinition Class

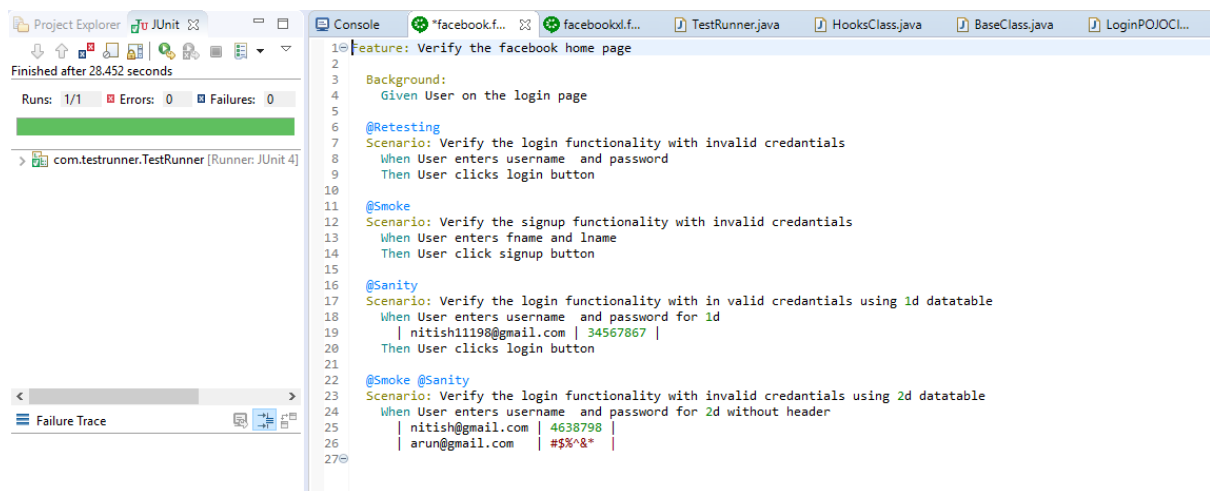
CUCUMBER



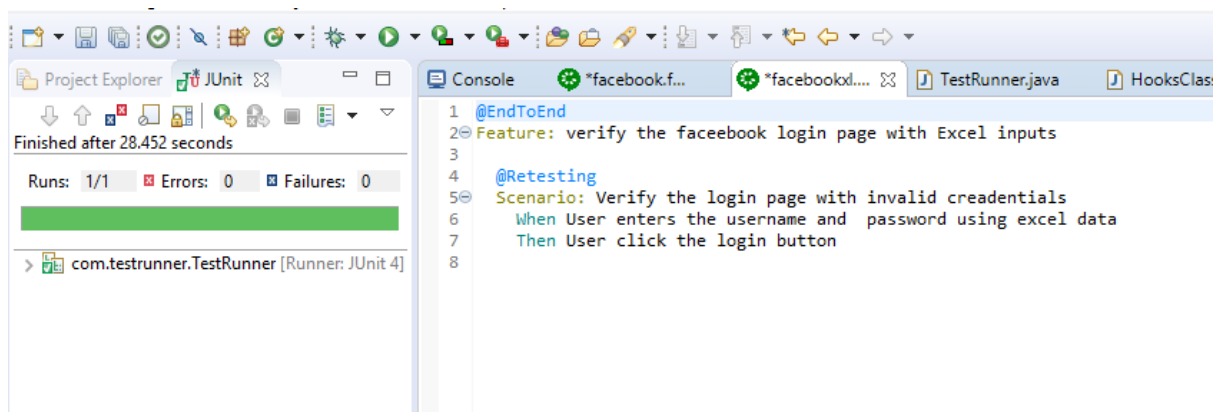
Output

Feature File Grouping

➤ tags={"@EndToEnd"}

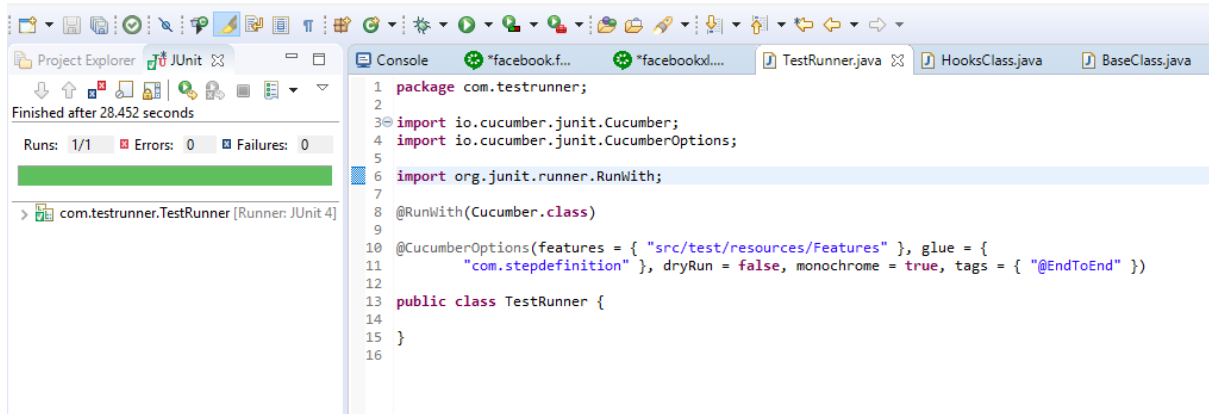


FeatureFile One

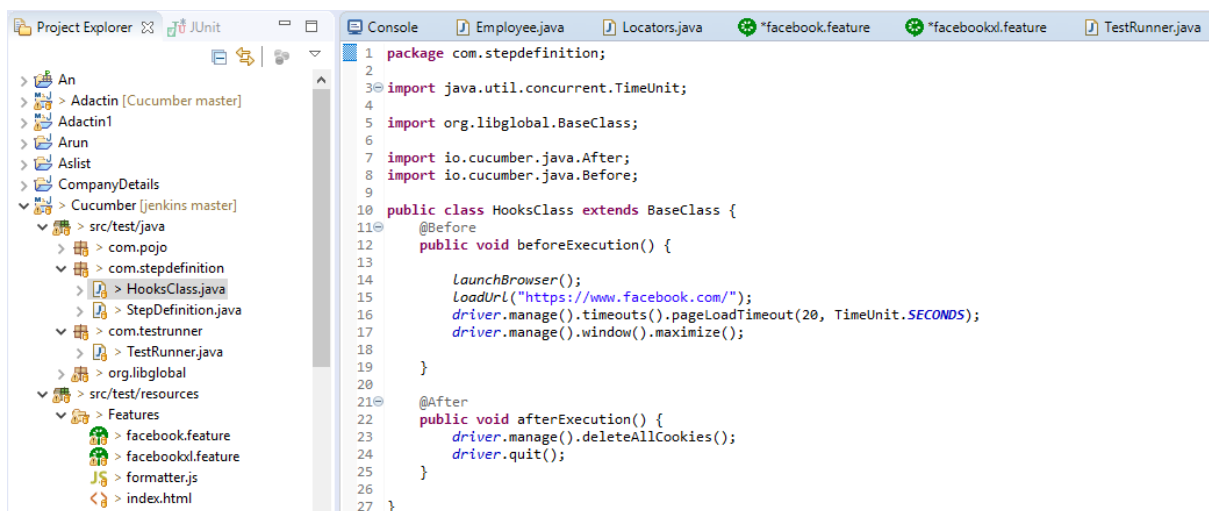


Feature File Two

CUCUMBER



TestRunnerClass



Hooks Class

```
package org.libglobal;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;

import org.apache.commons.io.FileUtils;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.DateUtil;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
```

CUCUMBER

```
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;

public class BaseClass {
    public static WebDriver driver;

    public static void launchBrowser() {
        System.setProperty("webdriver.chrome.driver",
            "C:\\\\Users\\k_sur\\Desktop\\Nitish\\cucumber
jenkins\\jenkins\\driver\\chromedriver.exe");
        driver = new ChromeDriver();
    }

    public static void maximizeWindow() {
        driver.manage().window().maximize();
    }

    public static void loadUrl(String url) {
        driver.get(url);
    }

    public static void type(WebElement e, String value) {
        e.sendKeys(value);
    }

    public static void btnClick(WebElement e) {
        e.click();
    }

    public static String getTittle() {
        String title = driver.getTitle();
        return title;
    }

    public static String getUrl() {
        String Url = driver.getCurrentUrl();
        return Url;
    }

    public static void quitBrowser() {
        driver.quit();
    }

    public static void selectByVisibleText(WebElement element, String text) {
        new Select(element).selectByVisibleText(text);
    }

    public static String getTextAttribute(WebElement element) {
        String attribute = element.getAttribute("value");
    }
}
```

CUCUMBER

```
        return attribute;
    }

    @SuppressWarnings({ "deprecation", "resource" })
    public static String getDataFromExcel(int row, int cell) throws IOException {
        String value = null;
        File loc = new File("C:\\Users\\k_sur\\eclipse-
workspace\\Cucumber\\Excel\\integration.xlsx");
        FileInputStream stream = new FileInputStream(loc);
        Workbook w = new XSSFWorkbook(stream);
        Sheet s = w.getSheet("Sheet1");
        Row r = s.getRow(row);
        Cell c = r.getCell(cell);

        int type = c.getCellType();

        if (type == 1) {

            value = c.getStringCellValue();

        } else if (type == 0) {
            if (DateUtil.isCellDateFormatted(c)) {
                Date dateCellValue = c.getDateCellValue();
                SimpleDateFormat sim = new SimpleDateFormat("dd-mm-yyyy");
                value = sim.format(dateCellValue);

            } else {

                double numericCellValue = c.getNumericCellValue();
                long l = (long) numericCellValue;
                value = String.valueOf(l);

            }

        }
        return value;
    }
}
```

BaseClass

```
package com.pojo;

import org.libglobal.BaseClass;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

public class LoginPOJOClass extends BaseClass {

    public LoginPOJOClass() {
        PageFactory.initElements(driver, this);
    }

    @FindBy(id = "email")
    private WebElement username;
```

CUCUMBER

```
@FindBy(id = "pass")
private WebElement password;

@FindBy(id = "loginbutton")
private WebElement login;

@FindBy(id = "u_0_m")
private WebElement fname;

@FindBy(id = "u_0_o")
private WebElement lname;

@FindBy(id = "u_0_13")
private WebElement signup;

public WebElement getUsername() {
    return username;
}

public WebElement getPassword() {
    return password;
}

public WebElement getLogin() {
    return login;
}

public WebElement getFname() {
    return fname;
}

public WebElement getLname() {
    return lname;
}

public WebElement getSignup() {
    return signup;
}
}
```

Pojo Class

```
package com.stepdefinition;

import java.io.IOException;
import java.util.List;
import org.libglobal.BaseClass;
import com.pojo.LoginPOJOClass;

import io.cucumber.java.en.*;

public class StepDefinition extends BaseClass {

    LoginPOJOClass l;

    @Given("User on the login page")
```

CUCUMBER

```
public void user_on_the_login_page() {

}

@When("User enters username and password")
public void user_enters_username_and_password() {
    l = new LoginPOJOClass();

    type(l.getUsername(), "nitish11198@gmail.com");
    type(l.getPassword(), "345678");

}

@Then("User clicks login button")

public void user_clicks_login_button() {
    l = new LoginPOJOClass();

    btnClick(l.getLogin());
}

@When("User enters fname and lname")
public void user_enters_fname_and_lname() {
    l = new LoginPOJOClass();

    type(l.getFname(), "nitish");
    type(l.getLname(), "kumar");

}

@Then("User click signup button")
public void user_click_signup_button() {
    l = new LoginPOJOClass();

    btnClick(l.getSignup());
}

@When("User enters username and password for {int}d")
public void user_enters_username_and_password_for_d(Integer int1,
io.cucumber.datatable.DataTable d) {
    List<String> li = d.asList();
    l = new LoginPOJOClass();

    type(l.getUsername(), li.get(0));
    type(l.getPassword(), li.get(1));
}

@When("User enters username and password for {int}d without header")
public void user_enters_username_and_password_for_d_without_header(Integer int1,
io.cucumber.datatable.DataTable d) {
    List<List<String>> list = d.asLists();
    l = new LoginPOJOClass();

    type(l.getUsername(), list.get(0).get(0));
    type(l.getPassword(), list.get(0).get(1));
}

@When("User enters the username and password")
```


CUCUMBER

```
public void user_enters_the_username_and_password() throws IOException {
    l = new LoginPOJOClass();
    type(l.getUsername(), getDataFromExcel(1, 1));
    type(l.getPassword(), getDataFromExcel(1, 1));
}

@Then("User clicks the login button")
public void user_clicks_the_login_button() {
    btnClick(l.getLogin());
}

@When("User enters the username and password using excel data")
public void user_enters_the_username_and_password_using_excel_data() throws IOException {
    l = new LoginPOJOClass();
    type(l.getUsername(), getDataFromExcel(1, 1));
    type(l.getPassword(), getDataFromExcel(1, 1));
}

@Then("User click the login button")
public void user_click_the_login_button() {
    l = new LoginPOJOClass();
    btnClick(l.getLogin());
}
}
```

Stepdfinition Class

The screenshot displays a browser window at the top showing the Facebook login page. The login form, including the email and password fields and the 'Log In' button, is highlighted with a red rectangle. Below the browser, an IDE window shows the test results. The 'TestRunner' console output indicates that the test passed, with a status of 'Runs: 1/1' and 'Errors: 0'. The console also shows the test scenario details, including the feature name 'verify the facebook login page with Excel inputs' and the step definition 'user_enters_the_username_and_password_using_excel_data()'. The IDE interface includes a Project Explorer on the left and a Console on the right.

Output

CUCUMBER