

## How to get the data from Excel sheet in Cucumber Feature file?

Getting from excel is not recommended for bdd framework. Actual purpose of bdd is without excel and stuff we can use the examples and data tables etc.

Gitlab code for Naveen – cucumber pom series.

[GitHub - naveenanimation20/LatestCucumber6WithPOM: Latest Cucumber6 BDD framework with POM\(Page Object Model\)](#)

You can download the Cucumber Eclipse Plugin:

[Cucumber-Eclipse > Update Site](#)

Cucumber Java Skeleton Repo::

[GitHub - cucumber/cucumber-java-skeleton: This is the simplest possible setup for Cucumber-JVM using Java.](#)

Latest eclipse plugin url:

[GitHub - cucumber/cucumber-eclipse-update-site-snapshot: Cucumber Eclipse Update Site Snapshots](#)

Full cucumber code repo-

[GitHub - naveenanimation20/Cucumber6LatestFeatures: Cucumber version 6 Latest Features](#)

Full cucumber code repo-

[GitHub - naveenanimation20/Cucumber6LatestFeatures: Cucumber version 6 Latest Features](#)

Full cucumber code repo-

[GitHub - naveenanimation20/Cucumber6LatestFeatures: Cucumber version 6 Latest Features](#)

Full code repo-

[GitHub - naveenanimation20/LatestCucumber6WithPOM: Latest Cucumber6 BDD framework with POM\(Page Object Model\)](#)

Full code repo-

[GitHub - naveenanimation20/LatestCucumber6WithPOM: Latest Cucumber6 BDD framework with POM\(Page Object Model\)](#)

Excel sheet prepared –

	A	B	C	D	E	F
1	subjectHeading	email	orderReference	message		
2	Customer service	<a href="mailto:naveen@gmail.com">naveen@gmail.com</a>	1	hi this is naveen here		
3	Webmaster	<a href="mailto:tom@gmail.com">tom@gmail.com</a>	2	hi this is tom here		
4						
5						

This is the page which we will automate-

Not secure | www.automationpractice.pl/index.php?controller=contact

!1... Acer nitro 5 ryzen ,s... Learning Selenium... DTS:X Ultra Not wo... DTS: X Ultra Not W... smar

Home Contact

## CUSTOMER SERVICE - CONTACT US

### SEND A MESSAGE

**Subject Heading**

Webmaster

If a technical problem occurs on this website

**Email address**

cucumbertesting1960@gmail.com

**Order reference**

-- Choose --

**Attach File**

No file selected Choose File

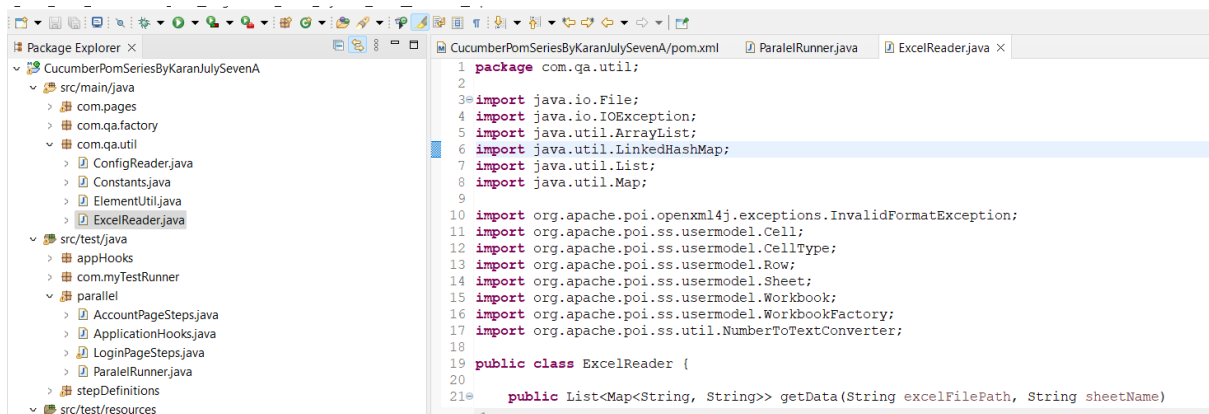
**Message**

Send >

Excel reader class-

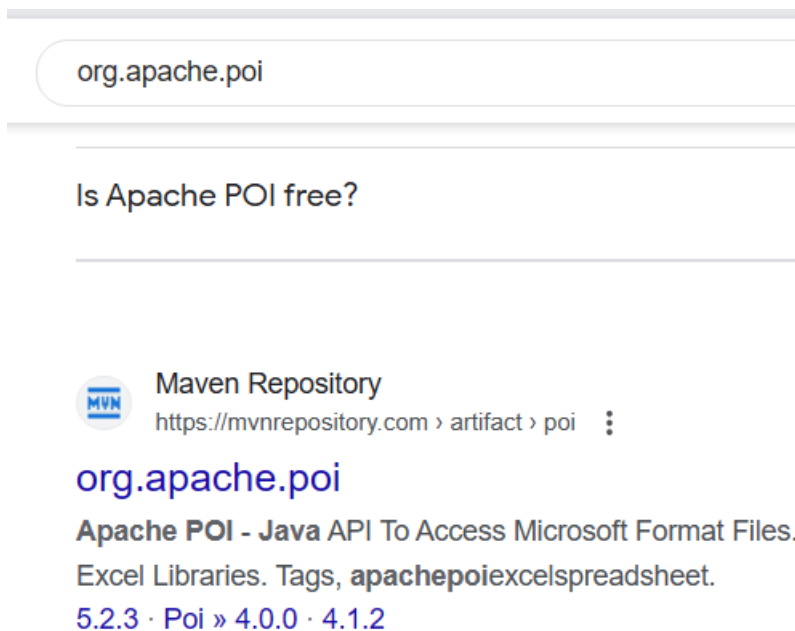
You can use anywhere. Its already built with all methods of excel. When time permits go through the methods.

This has been created by Naveen for working with excel.



We need to use apache poi api to work with excel-

Google search for it.



Click on the mvn link.

Select the latest version and paste in dependency.

[Home](#) » [org.apache.poi](#) » poi



## Apache POI Common

Apache POI - Java API To Access Microsoft Format Files

License	Apache 2.0
Categories	Excel Libraries
Tags	apache poi excel spreadsheet
Ranking	#198 in MvnRepository (See Top Artifacts) #1 in Excel Libraries
Used By	2,315 artifacts

Central (60)	Spring Plugins (2)	Spring Lib M (2)	BeDataDriven (6)	ImageJ Public (1)	Grails Core (1)	Alfresco 3rdParty (3)
Geomajas (1)	Alfresco (7)	Talend (3)	Aperture (14)	ICM (5)	Pellucid (7)	

	Version	Vulnerabilities	Repository	Usages	Date
5.2.x	5.2.3		Central	180	Sep 17, 2022
	5.2.2		Central	209	Mar 20, 2022
	5.2.1		Central	38	Mar 04, 2022
	5.2.0		Central	71	Jan 15, 2022
5.1.x	5.1.0		Central	85	Nov 02, 2021
5.0.x	5.0.0		Central	194	Jan 17, 2021
	4.1.2		Central	390	Feb 14, 2020

Add in pom file-

```

85
86      <!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->
87      <dependency>
88          <groupId>org.apache.poi</groupId>
89          <artifactId>poi</artifactId>
90          <version>4.1.2</version>
91      </dependency>
92

```

Another dependency we need to add-

Same version as poi should be added for this dependency.

×

[Images](#)
[Videos](#)
[Books](#)
[News](#)
[Maps](#)
[Flights](#)
[Finance](#)

---

About 238,000 results (0.37 seconds)

**Maven Repository**  
<https://mvnrepository.com/artifact/org.apache.poi-ooxml>

## Maven Repository: org.apache.poi » poi-ooxml

**Apache POI API Based On OPC and OOXML Schemas. Apache POI - Java API To Access Microsoft Format Files. License, Apache 2.0. Tags, poiapache.**

Used By: 2,444 artifacts

[4.1.2](#) · [5.2.2](#) · [5.2.3](#) · [5.0.0](#)

Select latest one and paste in pom.

[Home](#) » [org.apache.poi](#) » poi-ooxml

**Apache POI API Based On OPC and OOXML Schemas**  
 Apache POI - Java API To Access Microsoft Format Files

License	Apache 2.0
Tags	poi apache
Ranking	#186 in MvnRepository (See Top Artifacts)
Used By	2,444 artifacts

[Central \(49\)](#)
[Spring Plugins \(1\)](#)
[BeDataDriven \(4\)](#)
[Grails Core \(2\)](#)
[Alfresco 3rdParty \(3\)](#)
[EmergyaPub \(3\)](#)
[Geomajas \(1\)](#)

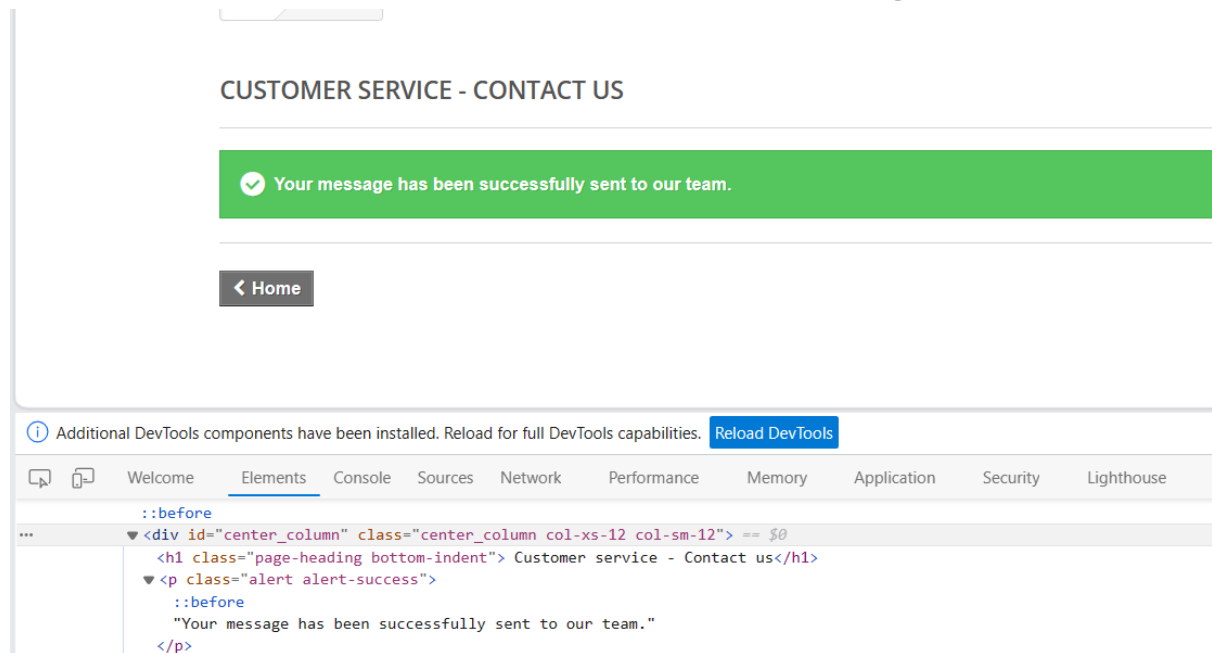
[Alfresco \(13\)](#)
[Talend \(3\)](#)
[Aperture \(12\)](#)
[ICM \(3\)](#)
[Pellucid \(6\)](#)

	Version	Vulnerabilities	Repository	Usages	Date
5.2.x	<a href="#">5.2.3</a>		<a href="#">Central</a>	236	Sep 17, 2022
	<a href="#">5.2.2</a>		<a href="#">Central</a>	271	Mar 20, 2022
	<a href="#">5.2.1</a>		<a href="#">Central</a>	42	Mar 04, 2022
	<a href="#">5.2.0</a>		<a href="#">Central</a>	85	Jan 15, 2022
5.1.x	<a href="#">5.1.0</a>		<a href="#">Central</a>	110	Nov 02, 2021
5.0.x	<a href="#">5.0.0</a>		<a href="#">Central</a>	217	Jan 17, 2021

```

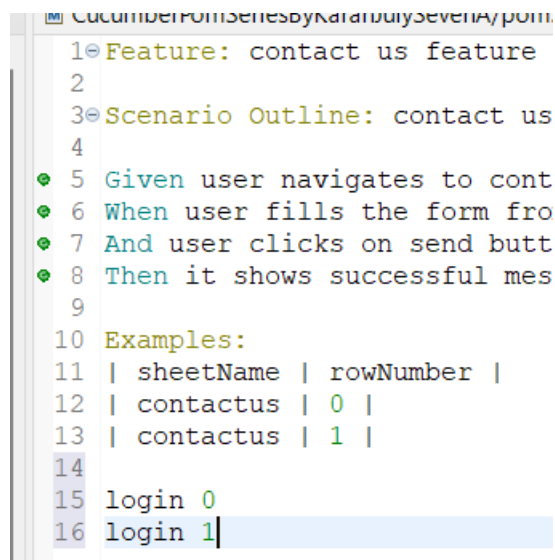
92
93      <!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->
94      <dependency>
95          <groupId>org.apache.poi</groupId>
96          <artifactId>poi-ooxml</artifactId>
97          <version>4.1.2</version>
98      </dependency>
99
100
101      </dependencies>
102
    
```

## How to write the css selector for the success message-



Div#center\_column p – <tagName#id> <childTag>

Tomorrow if you add another sheet for new page or application or scenario .. just add the sheet names as below-



Always close the excel first and then run the cases.

## Run the code for the excel-

Contact us feature page –

Feature: contact us feature

Scenario Outline: contact us scenario with different sets of data

Given user navigates to contact us page

When user fills the form from given sheetname "<sheetName>" and rownumber <rowNumber>

And user clicks on send button  
 Then it shows successful message "Your message has been successfully sent to our team."

#### Examples:

sheetName	rowNumber
contactus	0
contactus	1

Contact us page step def-  
 package parallel;

```
import java.io.IOException;
import java.util.List;
import java.util.Map;
```

```
import org.apache.poi.openxml4j.exceptions.InvalidFormatException;
import org.testng.Assert;
```

```
import com.pages.ContactUsPage;
import com.qa.factory.DriverFactory;
import com.qa.util.ExcelReader;
```

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

```
public class ContactUsPageSteps {
```

```
    private ContactUsPage contactUsPage=new ContactUsPage(DriverFactory.getDriver());
```

```
    @Given("user navigates to contact us page")
    public void user_navigates_to_contact_us_page() {
```

```
        DriverFactory.getDriver().get("http://www.automationpractice.pl/index.php?controller=contact");
    }
```

```
    @When("user fills the form from given sheetname {string} and rownumber {int}")
    public void user_fills_the_form_from_given_sheetname_and_rownumber(String
sheetName, Integer rowNumber)
```

```
        throws InvalidFormatException, IOException {
```

```
        //create object of excel reader to read the methods
```

```
        ExcelReader reader=new ExcelReader();
```

```
        //later on add this to config file and read the path of excel.
```

```
        List<Map<String, String>> testData= reader.getData
```

```
        ("E:\\Naveen Java
```

```
Training\\Cucumber\\CucumberPomSeriesByKaranJulySevenA\\automation.xlsx", sheetName);
```

```
        /**
```

```
        * capture all the row data
```

```
        */
```

```
        String heading=testData.get(rowNumber).get("subjectHeading");
```

```
        String email=testData.get(rowNumber).get("email");
```

```
        String orderReference=testData.get(rowNumber).get("orderReference");
```

```
        String message=testData.get(rowNumber).get("message");
```

```

        //now fill the contact us form
        contactUsPage.fillContactUsForm(heading, email, orderReference, message);
    }

    @When("user clicks on send button")
    public void user_clicks_on_send_button() {
        contactUsPage.clickSend();
    }

    @Then("it shows successful message {string}")
    public void it_shows_successful_message(String expectedSuccessMessage) {
        String actualSuccessMessage=contactUsPage.getSuccessMessage();
        Assert.assertEquals(actualSuccessMessage, expectedSuccessMessage);
    }
}

```

Contact us page class-  
package com.pages;

```

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.Select;

```

```

public class ContactUsPage {

```

```

    private WebDriver driver;

```

```

    private By subjectHeading=By.id("id_contact");
    private By email=By.id("email");
    private By orderRef=By.name("id_order");
    private By messageText=By.id("message");
    private By submitButton=By.id("submitMessage");
    private By successMessage=By.cssSelector("div#center_column p");

```

```

    public ContactUsPage(WebDriver driver) {
        this.driver=driver;
    }

```

```

    public String getContactUsPageTitle() {
        return driver.getTitle();
    }

```

//order ref and heading are select classes, so we use select class and create select class object and use  
//the select methods.

```

    public void fillContactUsForm(String heading, String emailId, String orderReference, String message) {
        Select select=new Select(driver.findElement(subjectHeading));
        select.selectByVisibleText(heading);
        driver.findElement(email).sendKeys(emailId);
    }

```



```

        driver.findElement(orderRef).sendKeys(orderReference);
        driver.findElement(messageText).sendKeys(message);
    }

    public void clickSend() {
        driver.findElement(submitButton).click();
    }

    public String getSuccessMessage() {
        return driver.findElement(successMessage).getText();
    }
}

Runner-
package parallel;

import org.testng.annotations.DataProvider;

import io.cucumber.junit.CucumberOptions;
import io.cucumber.testng.AbstractTestNGCucumberTests;

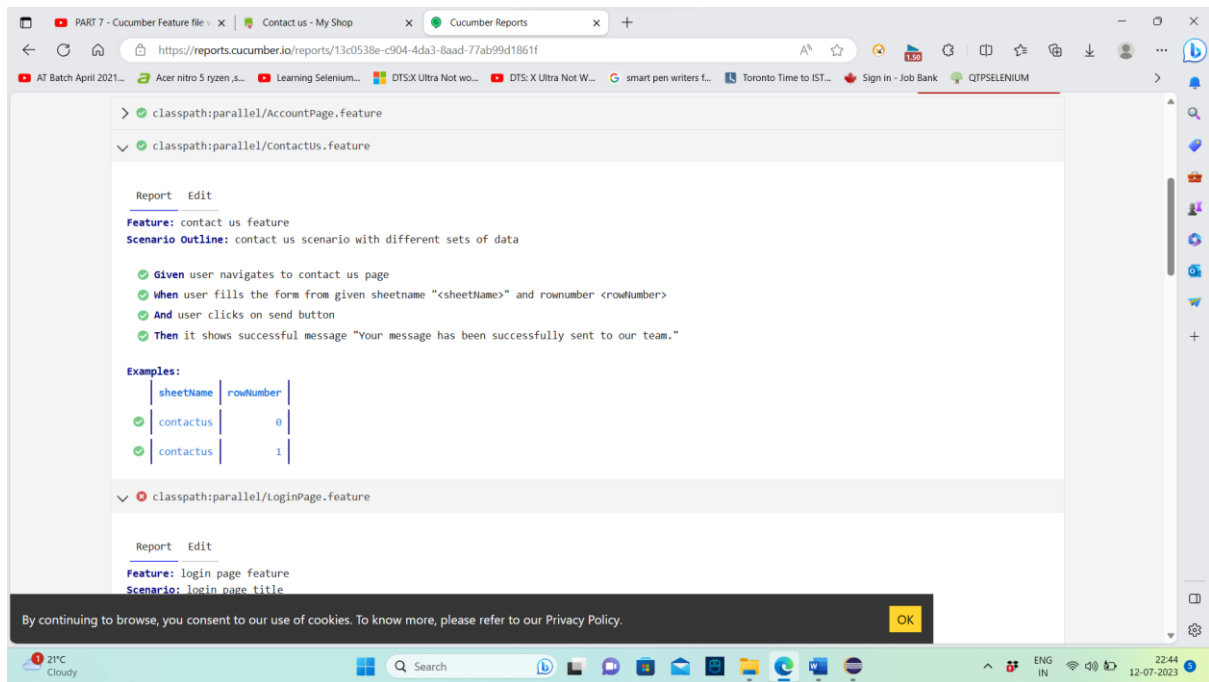
@CucumberOptions(features = {"src/test/resources/parallel/ContactUs.feature"},
glue = { "parallel" },
monochrome=true,
plugin = { "pretty",
           "com.aventstack.extentreports.cucumber.adapter.ExtentCucumberAdapter:",
           "timeline:test-output-thread/" }
)

public class ParallelRunner extends AbstractTestNGCucumberTests {

    /**
     * we need to override a method from abstracttestngcucumbertests class
     * parallel=true means run in parallel, else it will run one by one.
     *
     * @return 2d array containing all scenarios.
     */
    @Override
    @DataProvider(parallel = true)
    public Object[][] scenarios() {
        return super.scenarios();
    }
}

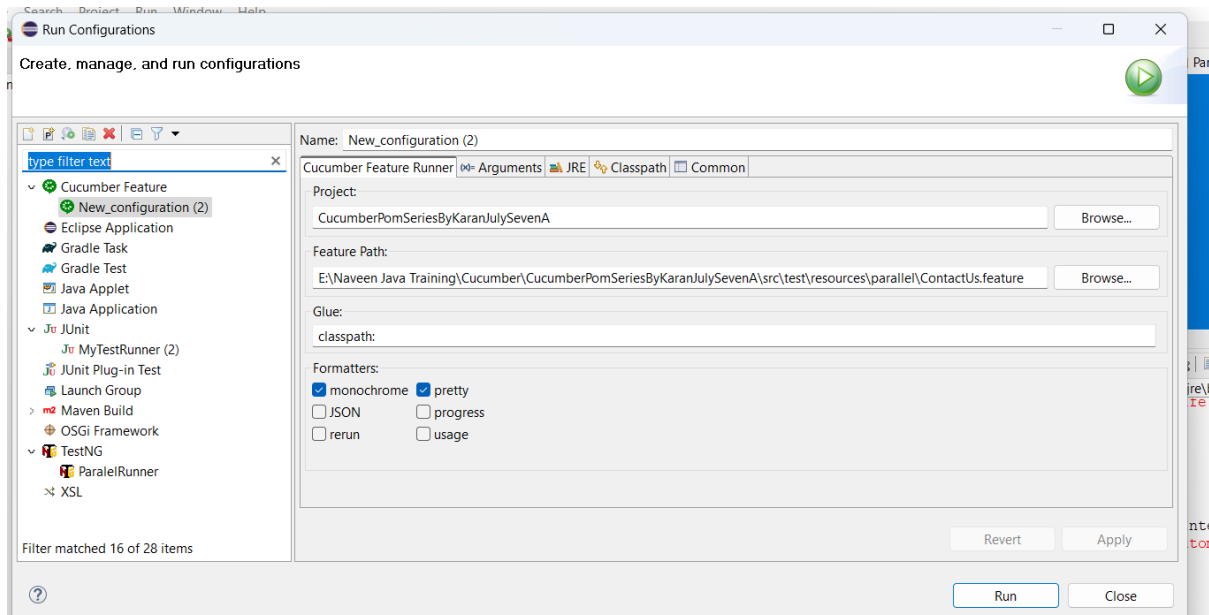
```

Cucumber report-



Sometimes all the feature files from the pack run even though we specify the file name. how to solve it.

Open run configuration. Edit and add wrong path in glue like `src/test/java/parallel/contactuspagesteps.java` and then run. It will throw error. Then remove and add only classpath: then run and it works. This also works sometimes and other times it doesn't work. apply and close.



Then we got the correct output.

```

Console x Results of running class ParallelRunner
<terminated> New_configuration (2) [Cucumber Feature] C:\Users\karan\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64.17.0.1
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver started successfully.
Jul 12, 2023 10:48:43 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Given user navigates to contact us page #
parallel.ContactUsPageSteps.user_navigates_to_contact_us_page() #
When user fills the form from given sheetname "contactus" and rownumber 1 #
parallel.ContactUsPageSteps.user_fills_the_form_from_given_sheetname_and_rownumber(java.lang.String) #
Cleaning up unclosed ZipFile for archive E:\Naveen Java Training\Cucumber\CucumberPomSeriesByKaranJ #
And user clicks on send button #
parallel.ContactUsPageSteps.user_clicks_on_send_button() #
Then it shows successful message "Your message has been successfully sent to our team." #
parallel.ContactUsPageSteps.it_shows_successful_message(java.lang.String) #

2 Scenarios (2 passed)
8 Steps (8 passed)
0m17.678s

View your Cucumber Report at:
https://reports.cucumber.io/reports/acee39b7-c839-415a-b567-f992d2ffd071

This report will self-destruct in 24h.
Keep reports forever: https://reports.cucumber.io/profile

```

#### Note-

Excel sheet code is written in such a way that row number starts from zero when we are writing our code to read from excel.

Added more rows in excel-

	A	B	C	D	E	F
1	subjectHeading	email	orderReference	message		
2	Customer service	<a href="mailto:naveen@gmail.com">naveen@gmail.com</a>	1	hi this is naveen here		
3	Webmaster	<a href="mailto:tom@gmail.com">tom@gmail.com</a>	2	hi this is tom here		
4	Customer service	<a href="mailto:naveen@gmail.com">naveen@gmail.com</a>	3	hi this is naveen here		
5	Webmaster	<a href="mailto:tom@gmail.com">tom@gmail.com</a>	4	hi this is tom here		
6	Customer service	<a href="mailto:naveen@gmail.com">naveen@gmail.com</a>	5	hi this is naveen here		
7	Webmaster	<a href="mailto:tom@gmail.com">tom@gmail.com</a>	6	hi this is tom here		
8						
9						
10						

We need to specify this in feature file also.

We have to specify the sheet name and the row numbers.

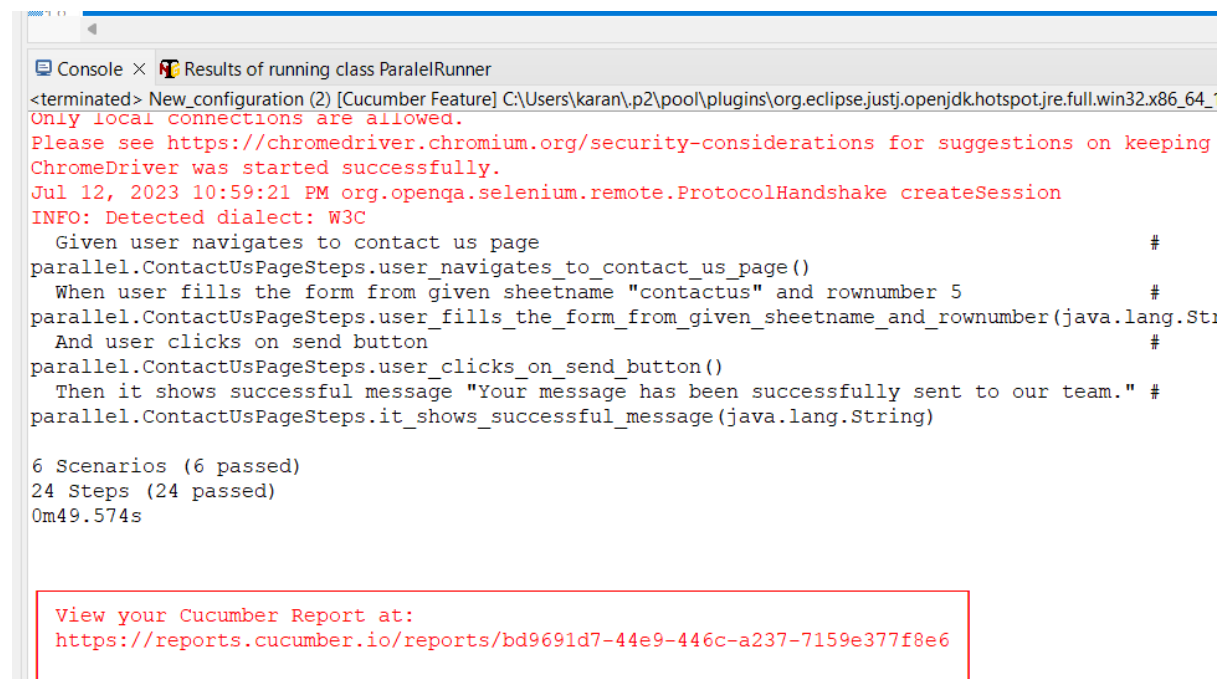
```

9
10 Examples:
11 | sheetName | rowNumber |
12 | contactus | 0 |
13 | contactus | 1 |
14 | contactus | 2 |
15 | contactus | 3 |
16 | contactus | 4 |
17 | contactus | 5 |

```

We get error as file cannot be processed if we keep the excel open when running our tests.

Run parallel runner and it should work fine.



```

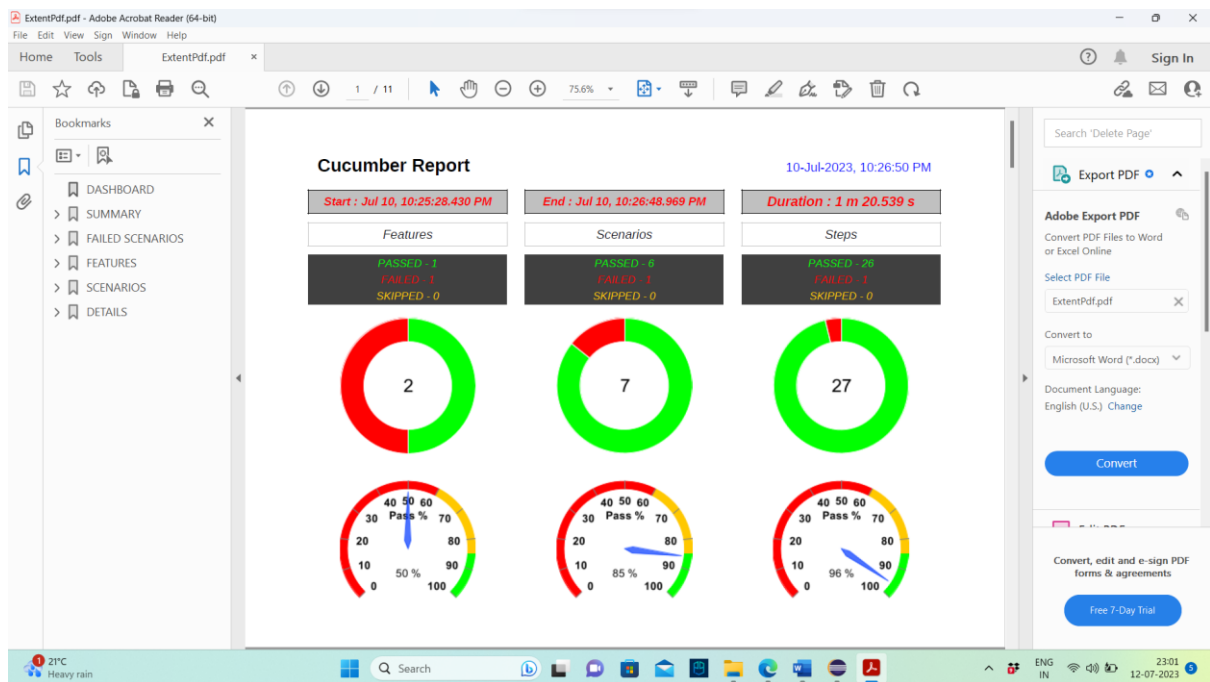
Console × Results of running class ParallelRunner
<terminated> New_configuration (2) [Cucumber Feature] C:\Users\karan\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping
ChromeDriver was started successfully.
Jul 12, 2023 10:59:21 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
    Given user navigates to contact us page #
parallel.ContactUsPageSteps.user_navigates_to_contact_us_page()
    When user fills the form from given sheetname "contactus" and rownumber 5 #
parallel.ContactUsPageSteps.user_fills_the_form_from_given_sheetname_and_rownumber(java.lang.Sti
    And user clicks on send button #
parallel.ContactUsPageSteps.user_clicks_on_send_button()
    Then it shows successful message "Your message has been successfully sent to our team." #
parallel.ContactUsPageSteps.it_shows_successful_message(java.lang.String)

6 Scenarios (6 passed)
24 Steps (24 passed)
0m49.574s

View your Cucumber Report at:
https://reports.cucumber.io/reports/bd9691d7-44e9-446c-a237-7159e377f8e6

```

Spark pdf report-



But the report shows sheet name and row number not the data inside it-

#	Step / Hook Text	Status
1	Given user navigates to contact us page	PASSED
2	When user fills the form from given sheetname "contactus" and row...	PASSED
3	And user clicks on send button	PASSED
4	Then it shows a successful message "Your message has been suc..."	PASSED

Same issue with index spark html report-

#	Step / Hook Text	Status
1	Given user navigates to contact us page	PASSED
2	When user fills the form from given sheetname "contactus" and rownumber 5	PASSED
3	And user clicks on send button	PASSED
4	Then it shows a successful message "Your message has been successfully sent to our team."	PASSED

Open cmd and go to the project location.

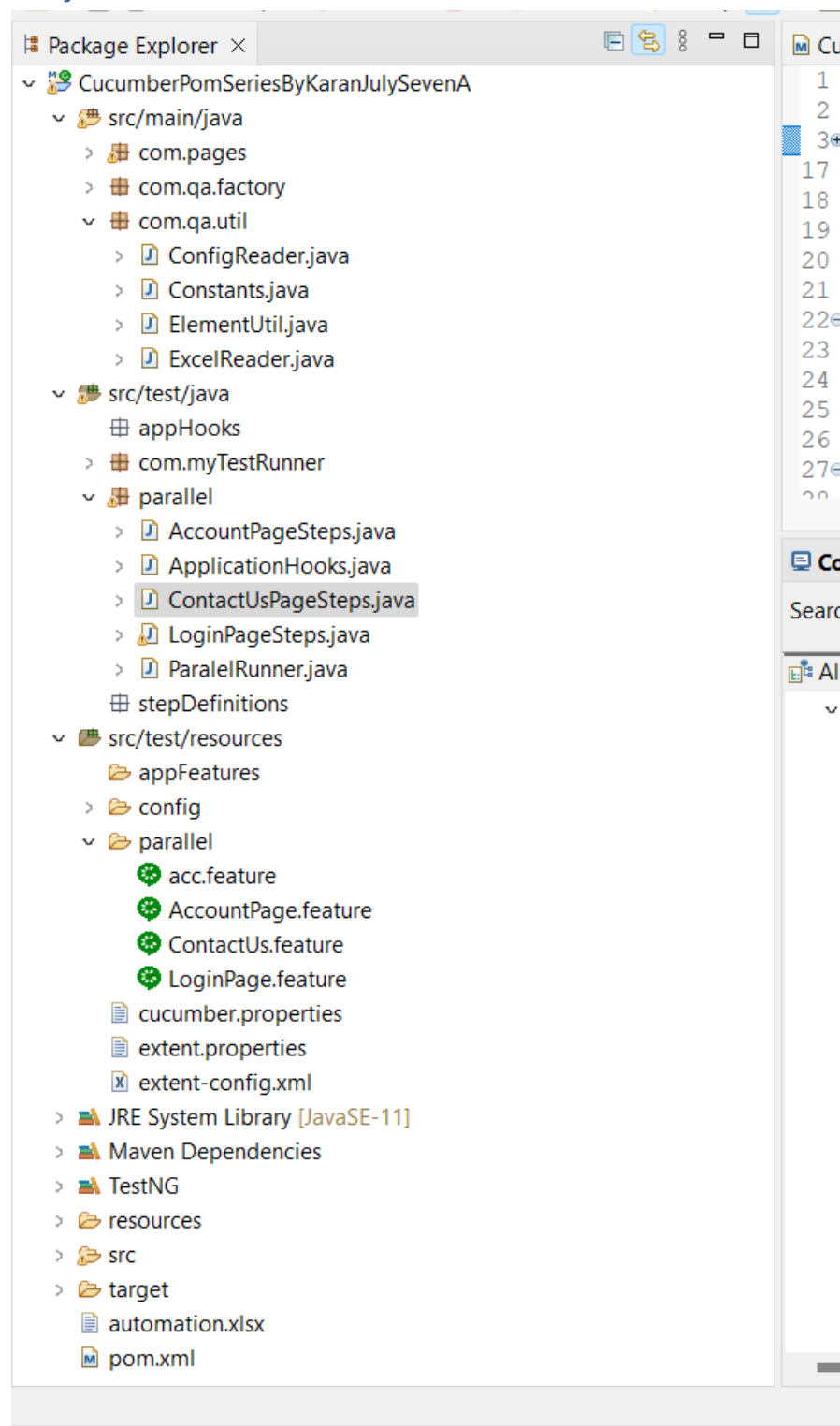
Run using the below command-

```
E:\Naveen Java Training\Cucumber\CucumberPomSeriesByKaranJulySevenA>mvn verify
```

Forgot password failing and rest passed-

```
ERROR] parallel.ParalelRunner.runScenario(parallel.ParalelRunner)
ERROR]   Run 1: ParalelRunner.runScenario » NoSuchElementException no such element: Unable to l
INFO]   Run 2: PASS
INFO]   Run 3: PASS
INFO]   Run 4: PASS
INFO]   Run 5: PASS
INFO]   Run 6: PASS
INFO]   Run 7: PASS
INFO]   Run 8: PASS
INFO]   Run 9: PASS
INFO]   Run 10: PASS
INFO]   Run 11: PASS
INFO]   Run 12: PASS
INFO]   Run 13: PASS
INFO]
INFO]
ERROR] Tests run: 1, Failures: 1, Errors: 0, Skipped: 0
INFO]
INFO] -----
INFO] BUILD FAILURE
INFO] -----
INFO] Total time: 36.125 s
INFO] Finished at: 2023-07-12T23:16:37+02:00
INFO] -----
WARNING]
```

## Project structure-



## Codes from this chapter-

Pom-

```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```

    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>CucumberPOMSeriesByKaranJulySeven</groupId>
    <artifactId>CucumberPOMSeriesByKaranJulySevenA</artifactId>
    <version>0.0.1-SNAPSHOT</version>

    <name>CucumberPOMSeries</name>
    <!-- FIXME change it to the project's website -->
    <url>http://www.example.com</url>

    <properties>

        <project.build.sourceEncoding>UTF-
8</project.build.sourceEncoding>
        <java.version>11</java.version>
        <junit.version>4.13.1</junit.version>
        <cucumber.version>6.9.0</cucumber.version>
        <maven.compiler.version>3.8.1</maven.compiler.version>
        <maven.surefire.version>2.22.2</maven.surefire.version>

    </properties>

    <dependencies>

        <dependency>
            <groupId>io.cucumber</groupId>
            <artifactId>cucumber-java</artifactId>
            <version>${cucumber.version}</version>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>io.cucumber</groupId>
            <artifactId>cucumber-junit</artifactId>
            <version>${cucumber.version}</version>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>${junit.version}</version>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>org.seleniumhq.selenium</groupId>
            <artifactId>selenium-java</artifactId>
            <version>3.141.59</version>
        </dependency>

        <dependency>
            <groupId>io.github.bonigarcia</groupId>
            <artifactId>webdrivermanager</artifactId>
            <version>4.2.2</version>
        </dependency>

    <!--

```



```

        https://mvnrepository.com/artifact/tech.grasshopper/extentreports-
cucumber6-adapter -->
        <dependency>
            <groupId>tech.grasshopper</groupId>
            <artifactId>extentreports-cucumber6-
adapter</artifactId>
            <version>2.17.0</version>
        </dependency>

        <dependency>
            <groupId>org.testng</groupId>
            <artifactId>testng</artifactId>
            <version>6.14.3</version>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>io.cucumber</groupId>
            <artifactId>cucumber-testng</artifactId>
            <version>${cucumber.version}</version>
            <scope>test</scope>
        </dependency>

        <!-- https://mvnrepository.com/artifact/org.apache.poi/poi --
>
        <dependency>
            <groupId>org.apache.poi</groupId>
            <artifactId>poi</artifactId>
            <version>4.1.2</version>
        </dependency>

        <!-- https://mvnrepository.com/artifact/org.apache.poi/poi-
ooxml -->
        <dependency>
            <groupId>org.apache.poi</groupId>
            <artifactId>poi-ooxml</artifactId>
            <version>4.1.2</version>
        </dependency>

    </dependencies>

    <build>
        <plugins>

            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>${maven.compiler.version}</version>
                <configuration>
                    <encoding>UTF-8</encoding>
                    <source>${java.version}</source>
                    <target>${java.version}</target>
                </configuration>
            </plugin>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-surefire-plugin</artifactId>

```

```

        <version>${maven.surefire.version}</version>
        <configuration>
            <!-- to run with jenkins and to run using
maven use the below includes -->
            <includes>

            <include>**/ParallelRunner.java</include>
            </includes>
            <parallel>methods</parallel>
            <!--<threadCount>4</threadCount>-->

            <useUnlimitedThreads>true</useUnlimitedThreads>
            </configuration>
        </plugin>

        <!-- <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-failsafe-plugin</artifactId>
            <version>3.1.2</version>
            <executions>
                <execution>
                    <goals>
                        <goal>integration-test</goal>
                    </goals>
                    <configuration> -->
                        <!-- UNCOMMENT - To add any
exclusions if required -->
                        <!-- <excludes>
<exclude>**/*IT*.java</exclude>
                        </excludes> -->
                        <!-- <includes> -->
                            <!--UNCOMMENT BELOW LINE
- To execute feature files with a single runner -->
                            <!--give path of runner
in the below step -->
                            <!--
                                <include>**/MyTestRunner.java</include> -->
                                    <!-- UNCOMMENT BELOW
LINE - To execute feature
                                files with multiple
runners -->
                                    <!--
<include>**/*Runner.java</include> -->
                                    <!--</includes> -->
                                    <!-- UNCOMMENT BELOW 3 LINES -
To execute using
                                parallel or combination
                                option -->
                                    <!-- we can write parallel with
classes also.-->
                                    <!-- <parallel>methods</parallel>
                                        <threadCount>10</threadCount>

                                <perCoreThreadCount>true</perCoreThreadCount> -->
                                    <!-- UNCOMMENT BELOW 3 LINES -
To execute using
                                forking or combination option -
->
                                    <!-- <forkCount>2</forkCount>
                                        <reuseForks>true</reuseForks>

```

```

        <reportsDirectory>${project.build.directory}/failsafe-
reports_${surefire.forkNumber}</reportsDirectory>-->
        <!--</configuration>
        </execution>
        </executions>
    </plugin>-->

</plugins>
</build>
</project>

```

Excel reader which is standard code written by Naveen-

```

package com.qa.util;

import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;

import org.apache.poi.openxml4j.exceptions.InvalidFormatException;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.CellType;
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.ss.usermodel.WorkbookFactory;
import org.apache.poi.ss.util.NumberToTextConverter;

public class ExcelReader {

    public List<Map<String, String>> getData(String excelFilePath,
String sheetName)
        throws InvalidFormatException, IOException {
        Sheet sheet = getSheetByName(excelFilePath, sheetName);
        return readSheet(sheet);
    }

    public List<Map<String, String>> getData(String excelFilePath, int
sheetNumber)
        throws InvalidFormatException, IOException {
        Sheet sheet = getSheetByIndex(excelFilePath, sheetNumber);
        return readSheet(sheet);
    }

    private Sheet getSheetByName(String excelFilePath, String
sheetName) throws IOException, InvalidFormatException {
        Sheet sheet = getWorkBook(excelFilePath).getSheet(sheetName);
        return sheet;
    }

    private Sheet getSheetByIndex(String excelFilePath, int
sheetNumber) throws IOException, InvalidFormatException {
        Sheet sheet =
getWorkBook(excelFilePath).getSheetAt(sheetNumber);
        return sheet;
    }
}

```

```

    private Workbook getWorkBook(String excelFilePath) throws
IOException, InvalidFormatException {
        return WorkbookFactory.create(new File(excelFilePath));
    }

    private List<Map<String, String>> readSheet(Sheet sheet) {
        Row row;
        int totalRow = sheet.getPhysicalNumberOfRows();
        List<Map<String, String>> excelRows = new
ArrayList<Map<String, String>>();
        int headerRowNumber = getHeaderRowNumber(sheet);
        if (headerRowNumber != -1) {
            int totalColumn =
sheet.getRow(headerRowNumber).getLastCellNum();
            int setCurrentRow = 1;
            for (int currentRow = setCurrentRow; currentRow <=
totalRow; currentRow++) {
                row = getRow(sheet, sheet.getFirstRowNum() +
currentRow);
                LinkedHashMap<String, String> columnMapdata = new
LinkedHashMap<String, String>();
                for (int currentColumn = 0; currentColumn <
totalColumn; currentColumn++) {
                    columnMapdata.putAll(getCellValue(sheet,
row, currentColumn));
                }
                excelRows.add(columnMapdata);
            }
        }
        return excelRows;
    }

    private int getHeaderRowNumber(Sheet sheet) {
        Row row;
        int totalRow = sheet.getLastRowNum();
        for (int currentRow = 0; currentRow <= totalRow + 1;
currentRow++) {
            row = getRow(sheet, currentRow);
            if (row != null) {
                int totalColumn = row.getLastCellNum();
                for (int currentColumn = 0; currentColumn <
totalColumn; currentColumn++) {
                    Cell cell;
                    cell = row.getCell(currentColumn,
Row.MissingCellPolicy.CREATE_NULL_AS_BLANK);
                    if (cell.getCellType() == CellType.STRING)
{
                        return row.getRowNum();
                    } else if (cell.getCellType() ==
CellType.NUMERIC) {
                        return row.getRowNum();
                    } else if (cell.getCellType() ==
CellType.BOOLEAN) {
                        return row.getRowNum();
                    } else if (cell.getCellType() ==
CellType.ERROR) {
                        return row.getRowNum();
                    }
                }
            }
        }
    }

```

```

    }
    }
    return (-1);
}

private Row getRow(Sheet sheet, int rowNum) {
    return sheet.getRow(rowNum);
}

private LinkedHashMap<String, String> getCellValue(Sheet sheet, Row
row, int currentColumn) {
    LinkedHashMap<String, String> columnMapdata = new
LinkedHashMap<String, String>();
    Cell cell;
    if (row == null) {
        if
(sheet.getRow(sheet.getFirstRowNum()).getCell(currentColumn,
Row.MissingCellPolicy.CREATE_NULL_AS_BLANK)
        .getCellStyle() != CellType.BLANK) {
            String columnHeaderName =
sheet.getRow(sheet.getFirstRowNum()).getCell(currentColumn)
                .getStringCellValue();
            columnMapdata.put(columnHeaderName, "");
        }
    } else {
        cell = row.getCell(currentColumn,
Row.MissingCellPolicy.CREATE_NULL_AS_BLANK);
        if (cell.getCellType() == CellType.STRING) {
            if (sheet.getRow(sheet.getFirstRowNum())
                .getCell(cell.getColumnIndex(),
Row.MissingCellPolicy.CREATE_NULL_AS_BLANK)
                .getCellStyle() != CellType.BLANK) {
                String columnHeaderName =
sheet.getRow(sheet.getFirstRowNum()).getCell(cell.getColumnIndex())
                    .getStringCellValue();
                columnMapdata.put(columnHeaderName,
cell.getStringCellValue());
            }
        } else if (cell.getCellType() == CellType.NUMERIC) {
            if (sheet.getRow(sheet.getFirstRowNum())
                .getCell(cell.getColumnIndex(),
Row.MissingCellPolicy.CREATE_NULL_AS_BLANK)
                .getCellStyle() != CellType.BLANK) {
                String columnHeaderName =
sheet.getRow(sheet.getFirstRowNum()).getCell(cell.getColumnIndex())
                    .getStringCellValue();
                columnMapdata.put(columnHeaderName,
NumberToTextConverter.toText(cell.getNumericCellValue()));
            }
        } else if (cell.getCellType() == CellType.BLANK) {
            if (sheet.getRow(sheet.getFirstRowNum())
                .getCell(cell.getColumnIndex(),
Row.MissingCellPolicy.CREATE_NULL_AS_BLANK)
                .getCellStyle() != CellType.BLANK) {
                String columnHeaderName =
sheet.getRow(sheet.getFirstRowNum()).getCell(cell.getColumnIndex())
                    .getStringCellValue();
                columnMapdata.put(columnHeaderName, "");
            }
        } else if (cell.getCellType() == CellType.BOOLEAN) {
            if (sheet.getRow(sheet.getFirstRowNum())

```

```

        .getCell(cell.getColumnIndex(),
Row.MissingCellPolicy.CREATE_NULL_AS_BLANK)
        .getCellStyle() != CellType.BLANK) {
            String columnHeaderName =
sheet.getRow(sheet.getFirstRowNum()).getCell(cell.getColumnIndex())
                .getStringCellValue();
            columnMapdata.put(columnHeaderName,
Boolean.toString(cell.getBooleanCellValue()));
        }
        } else if (cell.getCellType() == CellType.ERROR) {
            if (sheet.getRow(sheet.getFirstRowNum())
                .getCell(cell.getColumnIndex(),
Row.MissingCellPolicy.CREATE_NULL_AS_BLANK)
                .getCellStyle() != CellType.BLANK) {
                String columnHeaderName =
sheet.getRow(sheet.getFirstRowNum()).getCell(cell.getColumnIndex())
                    .getStringCellValue();
                columnMapdata.put(columnHeaderName,
Byte.toString(cell.getErrorCellValue()));
            }
        }
    }
    return columnMapdata;
}
}

```

#### Feature file-

**Feature:** contact us feature

**Scenario Outline:** contact us scenario with different sets of data

**Given** user navigates to contact us page

**When** user fills the form from given sheetname "<sheetName>" and rownumber <rowNumber>

**And** user clicks on send button

**Then** it shows successful message "Your message has been successfully sent to our team."

#### Examples:

sheetName	rowNumber
contactus	0
contactus	1
contactus	2
contactus	3
contactus	4
contactus	5

#### Step def-

package parallel;

import java.io.IOException;

import java.util.List;

import java.util.Map;

import org.apache.poi.openxml4j.exceptions.InvalidFormatException;

import org.testng.Assert;

import com.pages.ContactUsPage;

import com.qa.factory.DriverFactory;

import com.qa.util.ExcelReader;

```

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

public class ContactUsPageSteps {

    private ContactUsPage contactUsPage=new ContactUsPage(DriverFactory.getDriver());

    @Given("user navigates to contact us page")
    public void user_navigates_to_contact_us_page() {

DriverFactory.getDriver().get("http://www.automationpractice.pl/index.php?controller=contact");
    }

    @When("user fills the form from given sheetname {string} and rownumber {int}")
    public void user_fills_the_form_from_given_sheetname_and_rownumber(String
sheetName, Integer rowNumber)
        throws InvalidFormatException, IOException {
        //create object of excel reader to read the methods
        ExcelReader reader=new ExcelReader();
        //later on add this to config file and read the path of excel.
        List<Map<String, String>> testData= reader.getData
        ("E:\\Naveen Java
Training\\Cucumber\\CucumberPomSeriesByKaranJulySevenA\\automation.xlsx", sheetName);
        /**
        * capture all the row data
        */
        String heading=testData.get(rowNumber).get("subjectHeading");
        String email=testData.get(rowNumber).get("email");
        String orderReference=testData.get(rowNumber).get("orderReference");
        String message=testData.get(rowNumber).get("message");
        //now fill the contact us form
        contactUsPage.fillContactUsForm(heading, email, orderReference, message);
    }

    @When("user clicks on send button")
    public void user_clicks_on_send_button() {
        contactUsPage.clickSend();
    }

    @Then("it shows successful message {string}")
    public void it_shows_successful_message(String expectedSuccessMessage) {
        String actualSuccessMessage=contactUsPage.getSuccessMessage();
        Assert.assertEquals(actualSuccessMessage, expectedSuccessMessage);
    }

}

```

Page class-  
package com.pages;

```

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.Select;

public class ContactUsPage {

    private WebDriver driver;

    private By subjectHeading=By.id("id_contact");
    private By email=By.id("email");
    private By orderRef=By.name("id_order");
    private By messageText=By.id("message");
    private By submitButton=By.id("submitMessage");
    private By successMessage=By.cssSelector("div#center_column p");

    public ContactUsPage(WebDriver driver) {
        this.driver=driver;
    }

    public String getContactUsPageTitle() {
        return driver.getTitle();
    }

    //order ref and heading are select classes, so we use select class and create select class
    //the select methods.
    object and use

    public void fillContactUsForm(String heading, String emailId, String orderReference, String
message) {
        Select select=new Select(driver.findElement(subjectHeading));
        select.selectByVisibleText(heading);
        driver.findElement(email).sendKeys(emailId);
        driver.findElement(orderRef).sendKeys(orderReference);
        driver.findElement(messageText).sendKeys(message);
    }

    public void clickSend() {
        driver.findElement(submitButton).click();
    }

    public String getSuccessMessage() {
        return driver.findElement(successMessage).getText();
    }
}

```

```

Runner-
package parallel;

```

```

import org.testng.annotations.DataProvider;

import io.cucumber.junit.CucumberOptions;
import io.cucumber.testng.AbstractTestNGCucumberTests;

```



```

@CucumberOptions(features = {"src/test/resources/parallel/ContactUs.feature"},
glue = { "parallel" },
monochrome=true,
plugin = { "pretty",
           "com.aventstack.extentreports.cucumber.adapter.ExtentCucumberAdapter:",
           "timeline:test-output-thread/" }
)

public class ParallelRunner extends AbstractTestNGCucumberTests {

    /**
     * we need to override a method from abstracttestngcumbertests class
     * parallel=true means run in parallel, else it will run one by one.
     *
     * @return 2d array containing all scenarios.
     */
    @Override
    @DataProvider(parallel = true)
    public Object[][] scenarios() {
        return super.scenarios();
    }
}

```

Latest code available on GIT repo: [GitHub - naveenanimation20/LatestCucumber6WithPOM: Latest Cucumber6 BDD framework with POM\(Page Object Model\)](#)

I removed all the feature files from parallel folder and placed in another folder and it is now running only contact us page-

