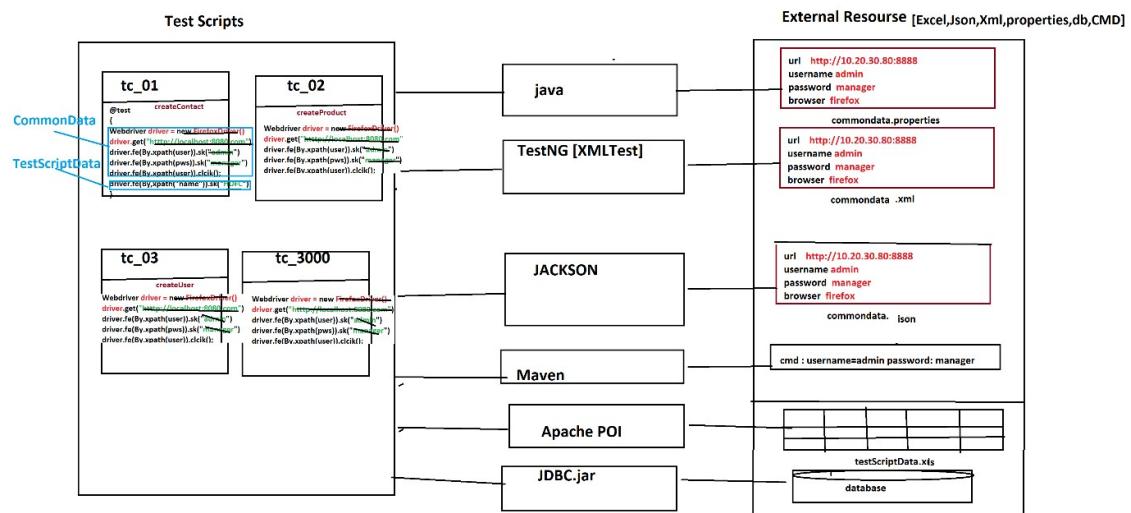


Data Driven testing

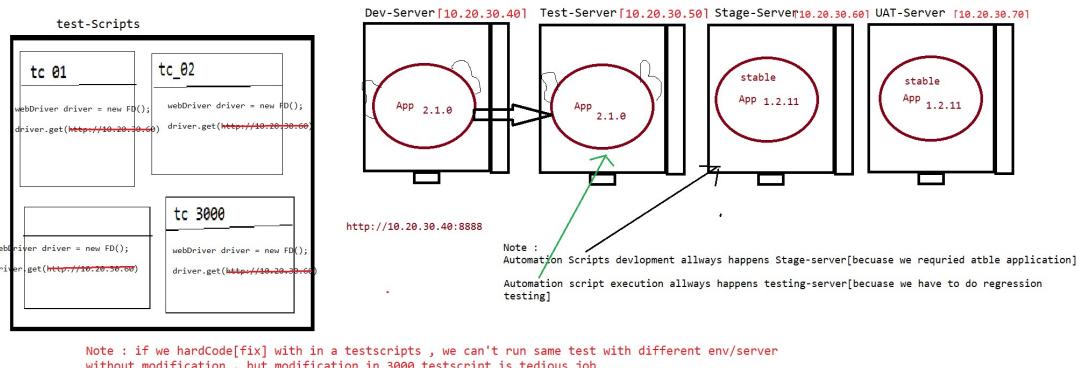
1. What is Data driven testing?

Read the data from external recourse & run the test is called Data driven testing (parameterization)



2. Why Data Driven testing?

As per the rule of the automation data shouldn't not hardcoded(fixed)with in a test scripts, because data modification & maintenance is tedious job when you want to run the test with different data, instead we should get the data from external resource like xlsx , .properties file , db , XML, JSON, CMD Line Data



3. What is Advantages of Data driven testing
 1. Maintenance of the test data is easy
 2. Modification of the test data in external recourse is easy
 3. Cross browser /platform testing is easy (means change the browser in property File)
 4. Running test scripts in different Environment is easy
 5. Running test scripts in different credentials is easy
 6. We can create the test data prior the Suite execution (we can also get the data from testDatateam)
 7. Rerunning same test Script with multiple time with different data is easy

Data driven testing from Properties File

1. What is Properties File?

Properties is java feature file where we can store the data in form of key & values pair,
Key & value data type should be always string.

```
url http://10.20.30.40:8888
browser firefox
username root
password manager
timeout 10
```

data.properties

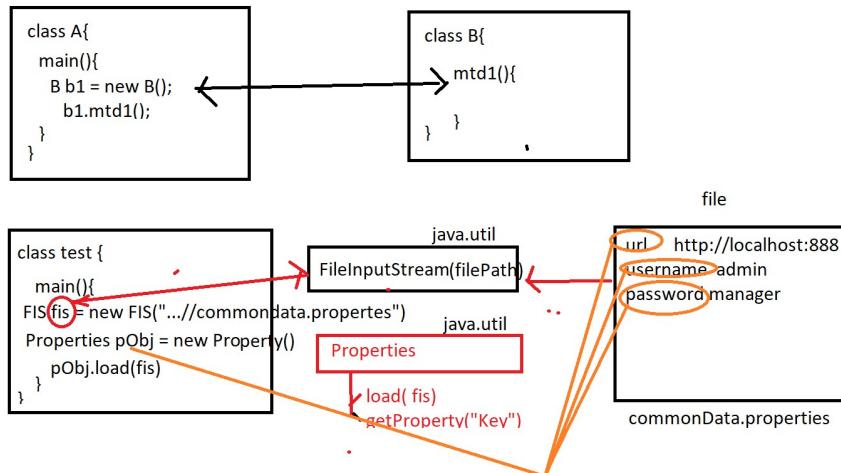
2. Why Property file ?

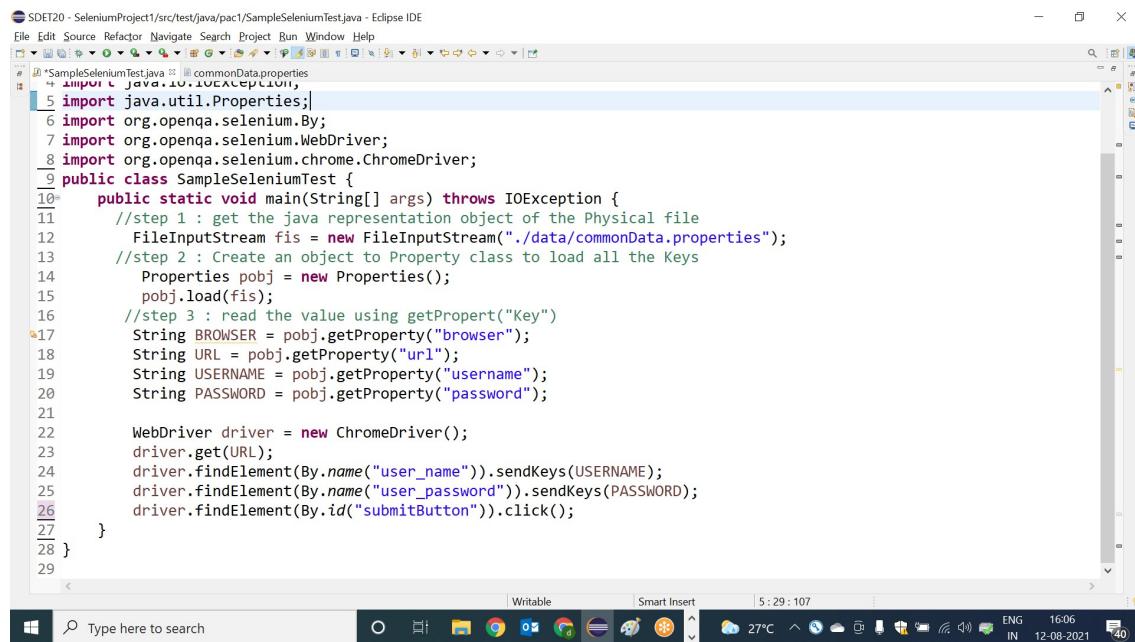
Property file is light weight & faster to read the data compare to any other file ,&
java as own Class to read the data from property

3. How to read data from properties File?

- Get the java representation Object of the Physical file using “FileInputStream”
- Create a Object of “Properties” class & load all the keys
- Read the data using getProperty(“Key”)

Note : properties file light weight & faster in execution compare to Excel



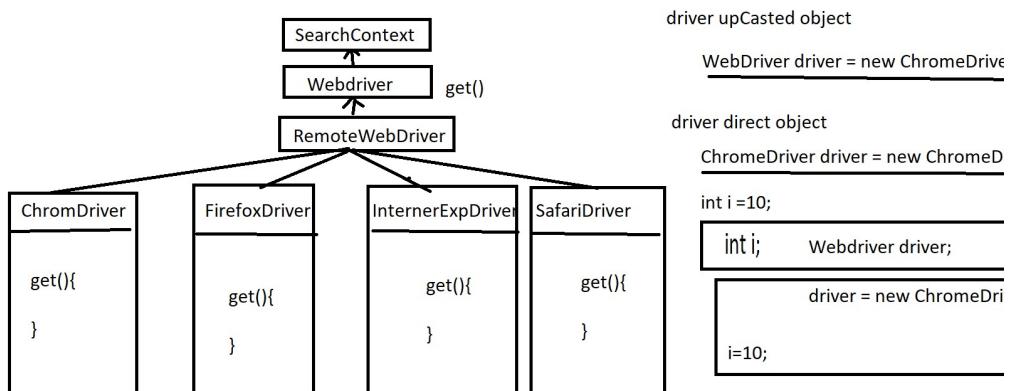


```

SDET20 - SeleniumProject1/src/test/java/pac1/SampleSeleniumTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
SampleSeleniumTest.java commonData.properties
1 import java.io.IOException;
2 import java.util.Properties;
3 import org.openqa.selenium.By;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.chrome.ChromeDriver;
6 public class SampleSeleniumTest {
7     public static void main(String[] args) throws IOException {
8         //step 1 : get the java representation object of the Physical file
9         FileInputStream fis = new FileInputStream("./data/commonData.properties");
10        //step 2 : Create an object to Property class to load all the Keys
11        Properties pobj = new Properties();
12        pobj.load(fis);
13        //step 3 : read the value using getProperty("Key")
14        String BROWSER = pobj.getProperty("browser");
15        String URL = pobj.getProperty("url");
16        String USERNAME = pobj.getProperty("username");
17        String PASSWORD = pobj.getProperty("password");
18
19        WebDriver driver = new ChromeDriver();
20        driver.get(URL);
21        driver.findElement(By.name("user_name")).sendKeys(USERNAME);
22        driver.findElement(By.name("user_password")).sendKeys(PASSWORD);
23        driver.findElement(By.id("submitButton")).click();
24    }
25 }

```

How to use Browser data in Seleniumtest



EG : best example for run time Polymorphism , driver object behave differently in run time

Assignment:

TestCase : CreateOrganization

step 1 : login
 step 2 : navigate to Organization module
 step 3 : click on "create Organization" Button
 step 4 : enter all the details & create new Organization
 step 5 : verify Organization name in header of the msg
 step 6 : logout

Note :

1. Any data should not be hardcoded
 2. test should able to run in different browser with minimal changes
 3. test script data should get it from EXCEL sheet
-

```
SDET_27 - AutoDeskSeleniumFramework/src/test/java/com/crm/autodesk/orgtest/CreateOrgTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
CreateOrgTest.java commonData.properties GetRandDomData.java testdata.xlsx
19 public class CreateOrgTest {
20     public static void main(String[] args) throws Throwable {
21         //read common data from properties file
22         FileInputStream fis = new FileInputStream("./data/commonData.properties");
23         Properties pobj = new Properties();
24         pobj.load(fis);
25         String URL = pobj.getProperty("url");
26         String USERNAME = pobj.getProperty("username");
27         String PASSWORD = pobj.getProperty("password");
28         String BROWSER = pobj.getProperty("browser");
29
30         //getRanDom Num
31         Random ran = new Random();
32         int ranDomNum = ran.nextInt(10000);
33
34         //read test data from Excel File
35         FileInputStream fis_e = new FileInputStream("./data/testdata.xlsx");
36         Workbook wb = WorkbookFactory.create(fis_e);
37         Sheet sh = wb.getSheet("org");
38         Row row = sh.getRow(1);
39         String orgName = row.getCell(2).getStringCellValue() + ranDomNum;
40
41
42         WebDriver driver;
43         if(BROWSER.equals("firefox")) {
44             driver = new FirefoxDriver();
45         }else if(BROWSER.equals("chrome")) {
46             driver = new ChromeDriver();
47         }else if(BROWSER.equals("ie")) {
48             driver = new InternetExplorerDriver();
49
50         }else {
51             driver = new ChromeDriver();
52         }
53         //step 1 : login
54         driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
55         driver.get(URL);
56         driver.findElement(By.name("user_name")).sendKeys(USERNAME);
57         driver.findElement(By.name("user_password")).sendKeys(PASSWORD);
58         driver.findElement(By.id("submitButton")).click();
59         // step 2 : navigate to Organization module
60         driver.findElement(By.linkText("Organizations")).click();
61         // step 3 : click on "create Organization" Button
62         driver.findElement(By.xpath("//img[@alt='Create Organization...']")).click();
63         // step 4 : enter all the details & create new Organization
64         driver.findElement(By.name("accountname")).sendKeys(orgName);
65         driver.findElement(By.xpath("//input[@title='Save [Alt+S]']")).click();
66         // step 5 : verify Organization name in header of the msg
67         String actSuc_msg = driver.findElement(By.className("dvHeaderText")).getText();
68         if(actSuc_msg.contains(orgName)) {
69             System.out.println("org is sucessfully created..PASS");
70         }else {
71             System.out.println("org is not created..FAIL");
72         }
73         //step 6 : logout
74         Actions act = new Actions(driver);
75         act.moveToElement(driver.findElement(By.xpath("//img[@src='themes/softed/images/user.PNG']"))).perform();
76         driver.findElement(By.linkText("Sign Out")).click();
77     }
}
```

```
SDET_27 - AutoDeskSeleniumFramework/src/test/java/com/crm/autodesk/orgtest/CreateOrgTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
CreateOrgTest.java commonData.properties GetRandDomData.java testdata.xlsx
76     }
77 }
```

Data driven testing from Excel File

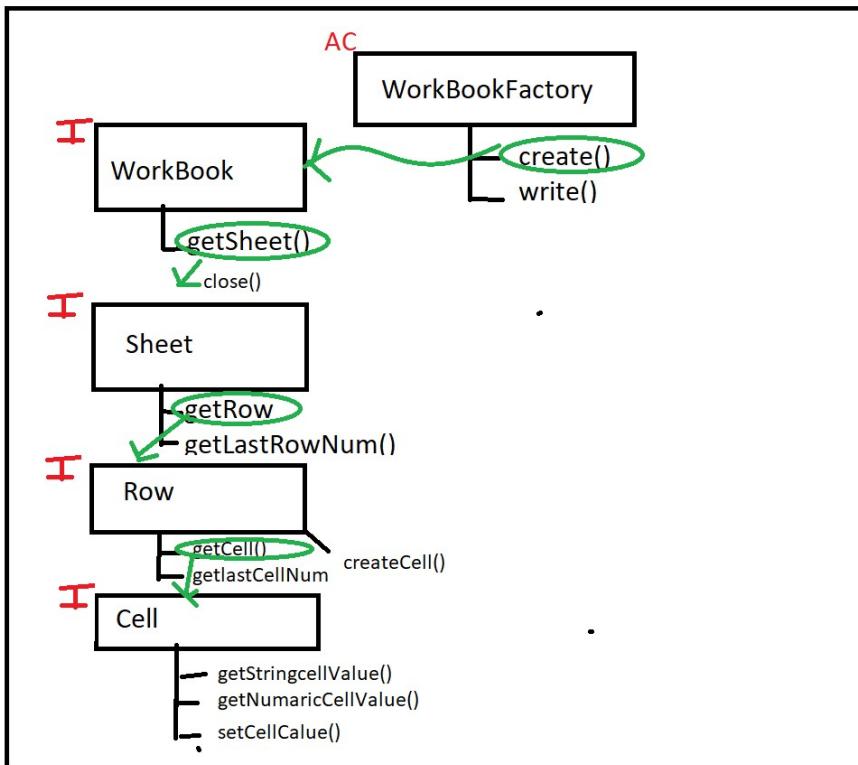
- Apache Poi is the open source libraries used to get & write data from all Microsoft documents like Excel , docx , pptetc
- In real time most the company preferred the keep the test script data in Excel, because data will be in well-organized manner , so that modification & maintenance is easier.

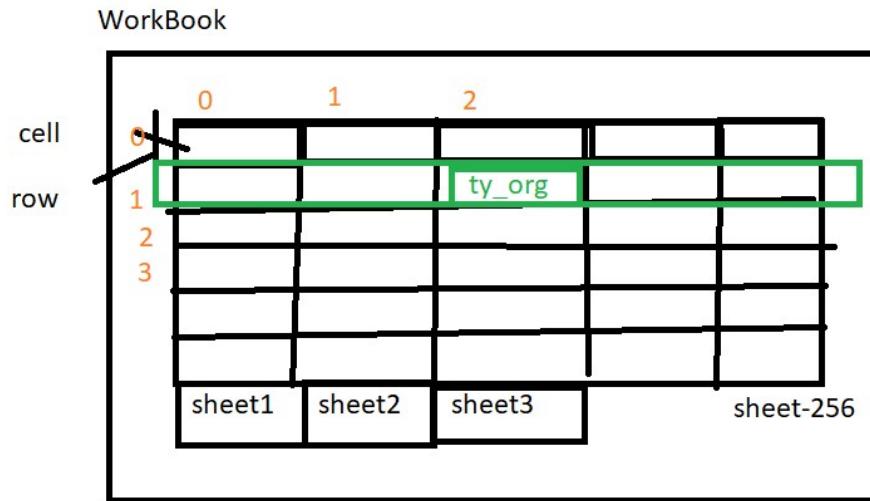
Installation steps:

1. Go the Maven Project
2. Edit POM.xml file
3. Go to <https://mavenrepository.com>
4. Search for apache-POI
5. Copy the dependency
6. Add dependency inside the <dependencies> in POM.xml

```
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi</artifactId>
    <version>4.0.0</version>
</dependency>
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml</artifactId>
    <version>4.0.0</version>
</dependency>
```

Class diagram of Apache POI





1. Program to read the data from WorkBook

```

SDT20 - SeleniumProject1/src/test/java/pac1/ReadDataFromExcel.java
File Edit Source Refactor Navigate Search Project Run Window Help
SampleSeleniumTest.java commonData.properties ReadDataFromExcelTest.java SeleniumProject1/pom.xml
1 import org.apache.poi.ss.usermodel.Cell;
2 import org.apache.poi.ss.usermodel.Row;
3 import org.apache.poi.ss.usermodel.Sheet;
4 import org.apache.poi.ss.usermodel.Workbook;
5 import org.apache.poi.ss.usermodel.WorkbookFactory;
6
7
8 public class ReadDataFromExcelTest {
9
10    public static void main(String[] args) throws Throwable {
11        FileInputStream fis = new FileInputStream("C:\\\\Users\\\\Deepak\\\\Desktop\\\\testScriptData.xlsx");
12        //Step 1 : Open WorkBook in read mode
13        Workbook wb = WorkbookFactory.create(fis);
14        //Step 2 : get the control of the Sheet-1
15        Sheet sh = wb.getSheet("Sheet1");
16        //Step 3 : get the control of the 1 st Row
17        Row row = sh.getRow(1);
18        //Step 4 : get the control of the 2 nd Cell & copy the data
19        Cell cel = row.getCell(2);
20        String data = cel.getStringCellValue();
21        System.out.println(data);
22        //Step 5 :close the WorkBook
23        wb.close();
24    }
25
26 }
27
28
29
30

```

The screenshot shows an IDE interface with the Java code for reading data from an Excel file. The code uses the Apache POI library to handle the workbook, specifically targeting the first sheet and the second cell of the second row to print its value ("ty_org").

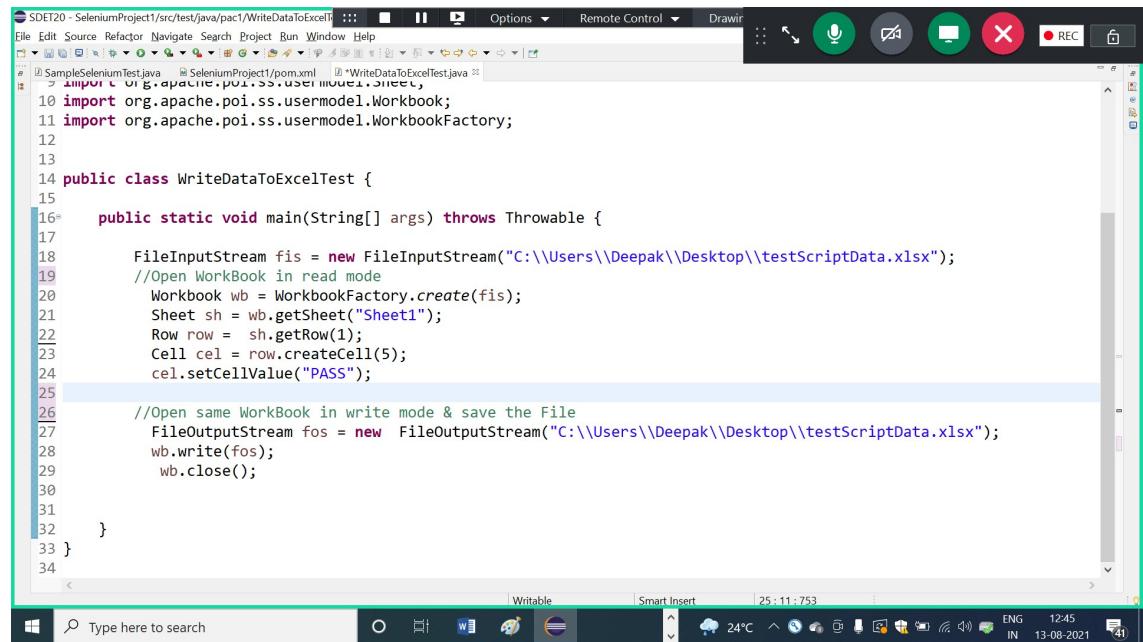
2. Program to read 1 st& 2 ndcoloum data from all the Rows

```
SDT20 - SeleniumProject1/src/test/java/pac1/ReadAllRowDataFr :: Options Remote Control Drawin
File Edit Source Refactor Navigate Search Project Run Window Help
SampleSeleniumTest.java commonData.properties ReadDataFromExcelTest.java SeleniumProject1/pom.xml *ReadAllRowDataFromExcelTest.java
1 package pac1;
2
3
4 import java.io.FileInputStream;
5
6
7 public class ReadAllRowDataFromExcelTest {
8
9     public static void main(String[] args) throws Throwable {
10         FileInputStream fis = new FileInputStream("C:\\Users\\Deepak\\Desktop\\testScriptData.xlsx");
11         Workbook wb = WorkbookFactory.create(fis);
12         Sheet sh = wb.getSheet("Sheet2");
13
14         //get the last used row count
15         int count = sh.getLastRowNum();
16
17         for(int i=1 ; i<count; i++) {
18             Row row = sh.getRow(i);
19             String firstColData = row.getCell(0).getStringCellValue();
20             String secondColData = row.getCell(1).getStringCellValue();
21             System.out.println(firstColData + "\t" + secondColData);
22         }
23
24     }
25
26
27 }
```

3. Write a program to find specific data from 1 stcolumn , if data is available then read &dispaly next cell data

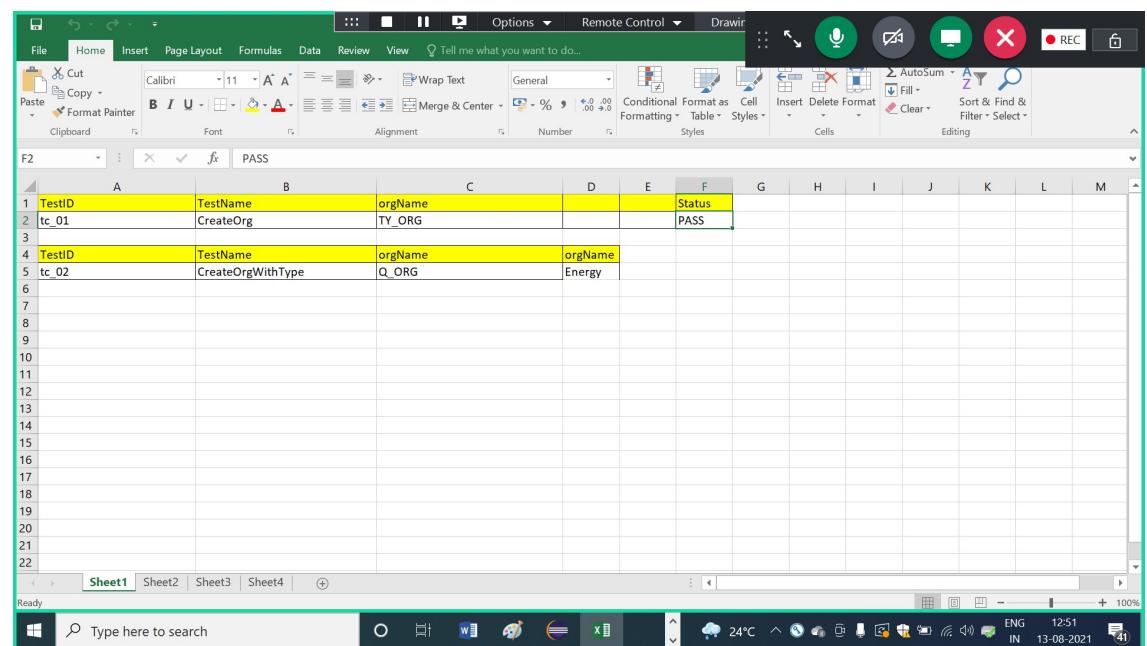
```
SDT20 - SeleniumProject1/src/test/java/pac1/ReadAllRowDataFr :: Options Remote Control Drawin
File Edit Source Refactor Navigate Search Project Run Window Help
SampleSeleniumTest.java commonData.properties ReadDataFromExcelTest.java SeleniumProject1/pom.xml *ReadAllRowDataFromExcelTest.java
13 public class ReadAllRowDataFromExcelTest {
14
15     public static void main(String[] args) throws Throwable {
16         FileInputStream fis = new FileInputStream("C:\\Users\\Deepak\\Desktop\\testScriptData.xlsx");
17         Workbook wb = WorkbookFactory.create(fis);
18         Sheet sh = wb.getSheet("Sheet2");
19
20         //get the last used row count
21         int count = sh.getLastRowNum();
22         String expectedData = "prhone-11";
23         for(int i=1 ; i<count; i++) {
24             Row row = sh.getRow(i);
25             String firstColData = row.getCell(0).getStringCellValue();
26
27             if(firstColData.equals(expectedData)) {
28
29                 String secondColData = row.getCell(1).getStringCellValue();
30                 System.out.println("available ==>" + secondColData);
31                 break;
32             }
33
34
35     }
36
37 }
```

4. Program to write data back to WorkBook/ Excel



The screenshot shows a Java code editor window titled "SDET20 - SeleniumProject1/src/test/java/pac1/WriteDataToExcelTest ::". The code is written in Java and uses the Apache POI library to interact with an Excel file. The code reads an Excel file, creates a new row with a single cell containing "PASS", and then writes the modified file back to disk.

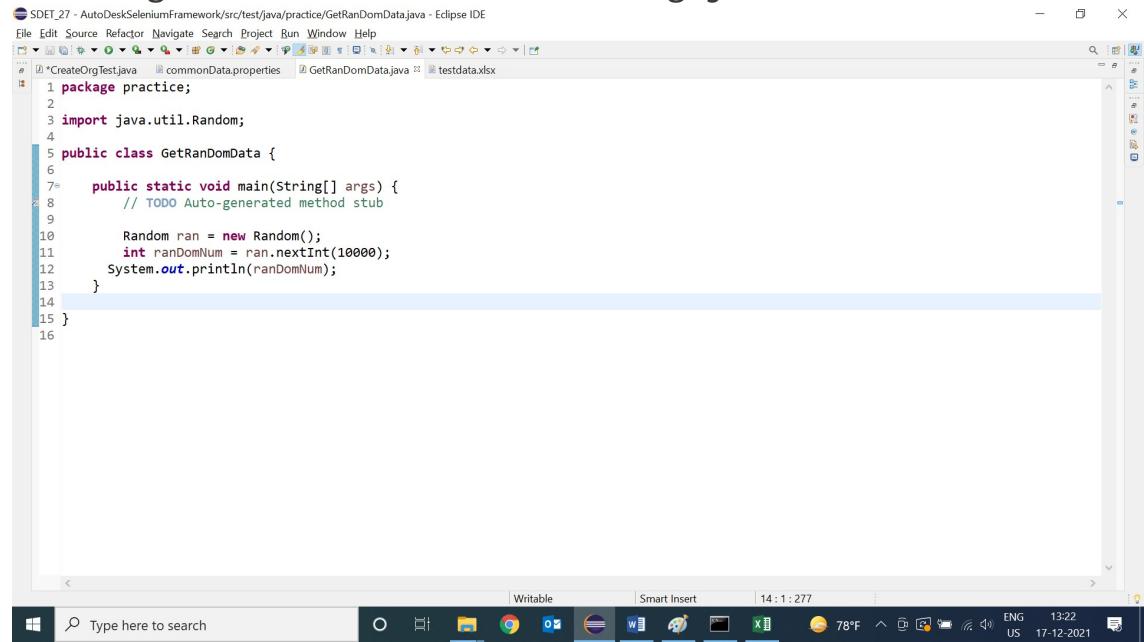
```
SampleSeleniumTest.java  SeleniumProject1/pom.xml  WriteDataToExcelTest.java
1 import org.g.apache.poi.ss.usermodel.Sheet;
2 import org.apache.poi.ss.usermodel.Workbook;
3 import org.apache.poi.ss.usermodel.WorkbookFactory;
4
5 public class WriteDataToExcelTest {
6
7     public static void main(String[] args) throws Throwable {
8
9         FileInputStream fis = new FileInputStream("C:\\\\Users\\\\Deepak\\\\Desktop\\\\testScriptData.xlsx");
10        //Open WorkBook in read mode
11        Workbook wb = WorkbookFactory.create(fis);
12        Sheet sh = wb.getSheet("Sheet1");
13        Row row = sh.getRow(1);
14        Cell cel = row.createCell(5);
15        cel.setCellValue("PASS");
16
17        //Open same WorkBook in write mode & save the File
18        FileOutputStream fos = new FileOutputStream("C:\\\\Users\\\\Deepak\\\\Desktop\\\\testScriptData.xlsx");
19        wb.write(fos);
20        wb.close();
21
22    }
23
24 }
25
```



The screenshot shows a Microsoft Excel spreadsheet titled "Sheet1". The data is organized into columns A through F. Rows 1 and 2 contain test cases, while rows 4 and 5 contain test cases with specific org types. The "Status" column consistently shows "PASS" for all entries.

A	B	C	D	E	F
1 TestID	TestName	orgName			Status
tc_01	CreateOrg	TY_ORG			PASS
4 TestID	TestName	orgName	orgName		
5 tc_02	CreateOrgWithType	Q_ORG	Energy		
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					

How to generate random number using java



The screenshot shows the Eclipse IDE interface with a Java file named 'GetRandDomData.java' open. The code generates a random integer between 0 and 10000 and prints it to the console.

```
package practice;
import java.util.Random;
public class GetRandDomData {
    public static void main(String[] args) {
        Random ran = new Random();
        int randomNum = ran.nextInt(10000);
        System.out.println(randomNum);
    }
}
```

Data-Driven Testing from CommandLine – using Maven Parameters

Why Java Main method argument is String array ?

Because its always accept String array of data from the commandline

4. How to get Maven Parameters(data like url , username , password) from the CMD line

1. Download maven Command line plugin & set Environment variables

- Intsllation steps
1. go to google
 2. search download maven
 3. click on first link
 4. click on " apache-maven-3.6.3-bin.zip" beside "Binary zip archi"
 5. download -ZIP & extract
 6. get inside the folder & copy the "bin" path location
 7. Go to Local system Env varibale window & set below path

Environment Variables

User variables for Deepak

Variable	Value
JAVA_HOME	C:\Program Files\Java\jdk1.8.0_161
M2_HOME	D:\apache-maven-3.5.2-bin\apache-maven-3.5.2

Environment Variables

User variables for Deepak

Variable	Value
JAVA_HOME	C:\Program Files\Java\jdk1.8.0_161
M2_HOME	D:\apache-maven-3.5.2-bin\apache-maven-3.5.2
OneDrive	C:\Users\Deepak\OneDrive
Path	C:\Users\Deepak\AppData\Local\Programs\Python\Python38-32\Scripts\;C:\Users\Deepak\PyCharm\bin\python\;C:\Program Files\JetBrains\PyCharm 2020.2.1\bin;
PyCharm	C:\Program Files\JetBrains\PyCharm Community Edition 2020.2.1\bin;
PyCharm Community Edition	C:\Program Files\JetBrains\PyCharm Community Edition 2020.2.1\bundles\PyCharmCommunityEdition\bin;
TEMP	C:\Users\Deepak\AppData\Local\Temp

Edit environment variable

C:\Program Files (x86)\Common Files\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\Java\jdk1.8.0_161\bin
D:\apache-maven-3.5.2-bin\apache-maven-3.5.2\bin
C:\Users\Deepak\AppData\Local\Android\Sdk\platform-tools
C:\Users\Deepak\AppData\Local\Android\Sdk\tools

System variables

Variable	Value
ANDROID_HOME	C:\Users\Deepak\AppData\Local\Android\Sdk
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\DriverData
NUMBER_OF_PROCESSORS	8
OS	Windows_NT
Path	C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\Program Files\Java\jdk1.8.0_161\bin;D:\apache-maven-3.5.2-bin\apache-maven-3.5.2\bin;C:\Users\Deepak\AppData\Local\Android\Sdk\platform-tools;C:\Users\Deepak\AppData\Local\Android\Sdk\tools
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH

8. go to command line , & execute below command to check maven installation

```
C:\Users\Deepak>mvn -version
Apache Maven 3.5.2 (138edd61fd100ec658bfa2d307c43b76940a5d7d; 2017-10-18T13:28:13+05:30)
Maven home: D:\apache-maven-3.5.2-bin\apache-maven-3.5.2\bin\..
Java version: 1.8.0_161, vendor: Oracle Corporation
Java home: C:\Program Files\Java\jdk1.8.0_161\jre
Default locale: en_IN, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
```

2. Create a Maven Project in Eclipse

3. Add testNG dependency in POM.xml

4. Create TestNg class

```
public class SampleTest {  
  
    @Test  
    Run | Debug  
    public void createContactTest() {  
        System.out.println("execute createContactTest");  
        String URL = System.getProperty("url");  
        String BROWSER = System.getProperty("browser");  
        String USERNAME = System.getProperty("username");  
  
        System.out.println(URL);  
        System.out.println(BROWSER);  
        System.out.println(USERNAME);  
  
    }  
}
```

5. Copy the location of the Maven Project

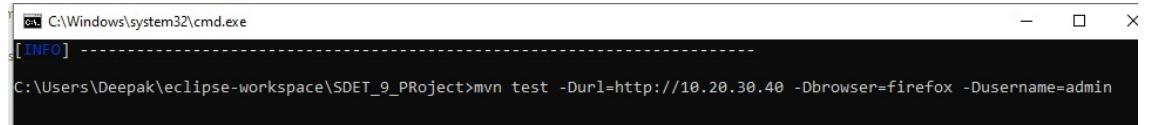
Project=>RightClick=>Properties=> Recourse=> copy the => location

6. Go to Cmd line

7. Go to the location of the Project

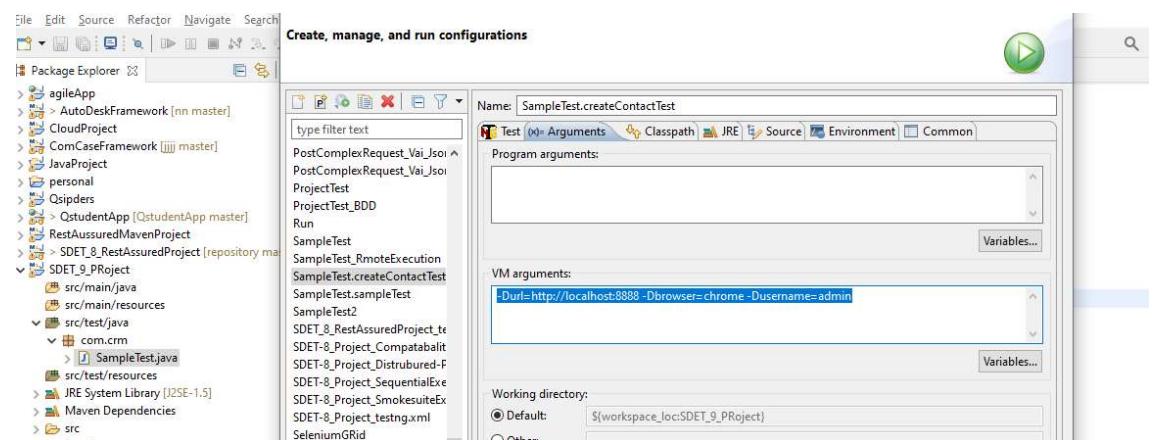
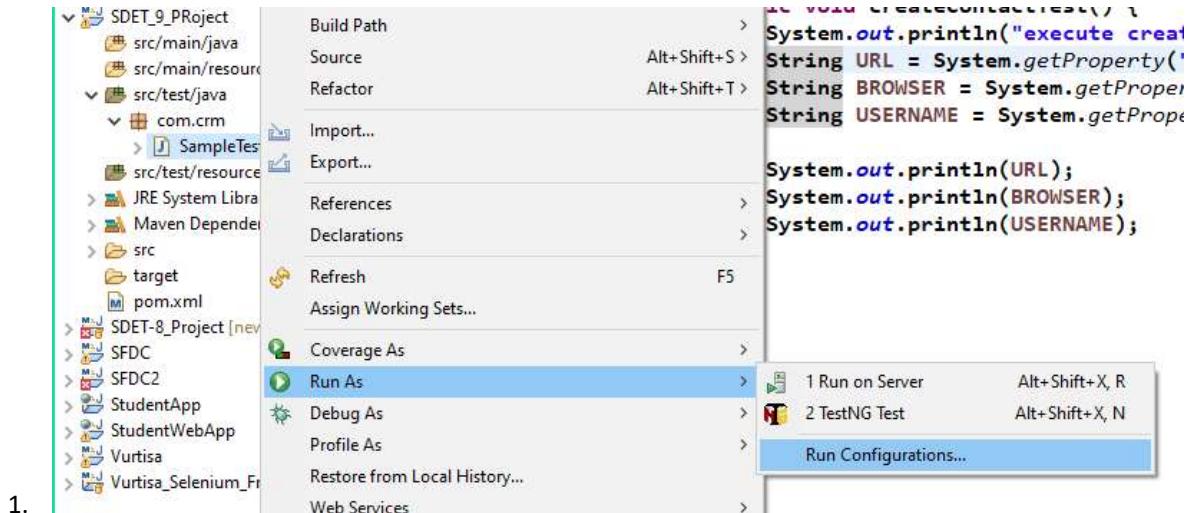
Cmd>cd C:\Users\Deepak\eclipse-workspace\SDT_9_Project

8. Execute below Command



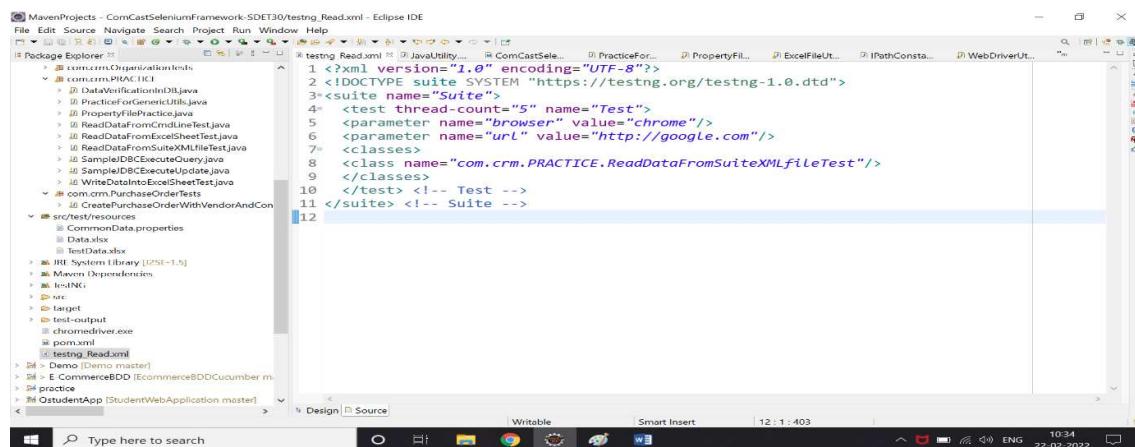
The screenshot shows a Windows Command Prompt window titled 'cmd C:\Windows\system32\cmd.exe'. The window contains the following text:
[INFO] -----
C:\Users\Deepak\eclipse-workspace\SDT_9_Project>mvn test -Durl=http://10.20.30.40 -Dbrowser=firefox -Dusername=admin

5. How to get Maven Parameters(data like url , username , password) from the Eclipse

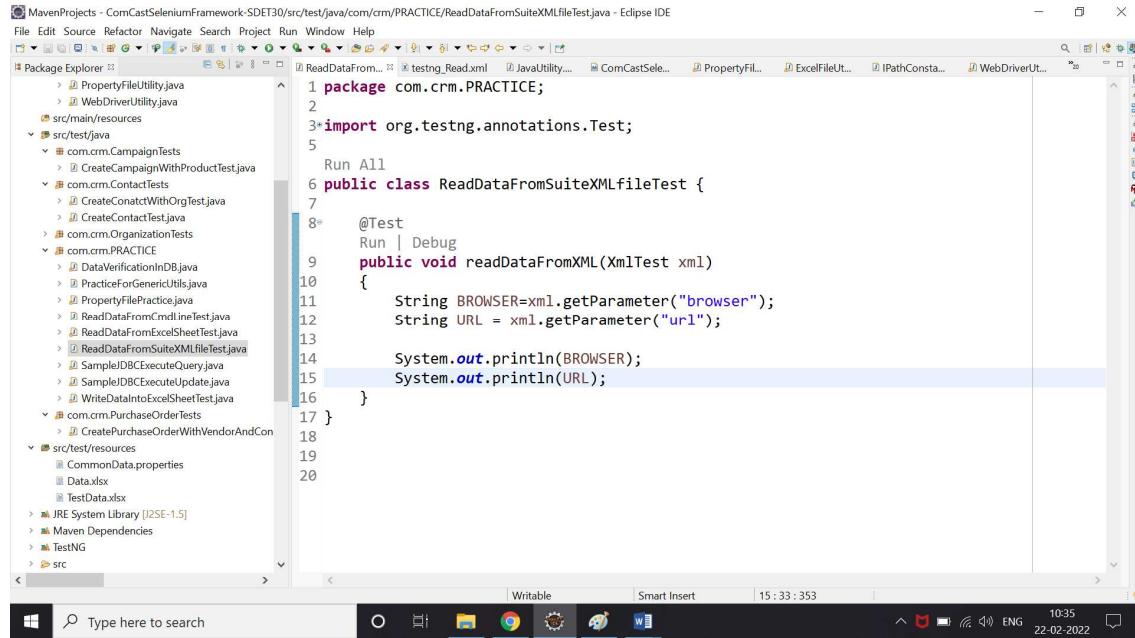


Data - Driven testing from Suite xml file

->Use parameter tag in suite xml file



→ Using `xmlTest` parameterise the method to read from suite xml file to the test script with the help of `getParameter()`.



The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** MavenProjects - ComCastSeleniumFramework-SDET30/src/test/java/com/crm/PRACTICE/ReadDataFromSuiteXMLfileTest.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar icons.
- Package Explorer:** Shows the project structure with packages like com.crm.CampaignTests, com.crm.ContactTests, com.crm.OrganizationTests, and com.crm.PRACTICE, along with various test classes.
- Editor:** Displays the Java code for `ReadDataFromSuiteXMLfileTest`. The code uses `xmlTest` annotations to read parameters from a suite XML file.

```
1 package com.crm.PRACTICE;
2
3 import org.testng.annotations.Test;
4
5 Run All
6 public class ReadDataFromSuiteXMLfileTest {
7
8     @Test
9     public void readDataFromXML(XmlTest xml) {
10         String BROWSER=xml.getParameter("browser");
11         String URL = xml.getParameter("url");
12
13         System.out.println(BROWSER);
14         System.out.println(URL);
15     }
16 }
17 }
```

- Bottom Status Bar:** Writable, Smart Insert, 15 : 33 : 353, 10:35, ENG, 22-02-2022.