Here we will follow a simple step by step process to Implement Excel with TestNg Data Provider.

**Step 1:** Create a test case of Login Application with TestNG Data Provider.

**Step 2:**  Create a Test Data sheet.

**Step 3:** Create functions to Open & Read data from Excel

**Step 4:** Create a TestNg test case for accepting data from Excel using Data Provider.

**Step 5:** Run the test against the Test Case name in the Test Data file.

**Step 1: Create a test case of LogIn Application with TestNG Data Provider**

1) Create a TestNG class ‘DataProviderTest’ by Pressing Ctrl+N , select ‘Create TestNG Class’ under TestNG category and Under Annotations, check ‘DataProvider’ and click Finish.

2) By default, the DataProvider name is ‘dp’, change it to ‘Authentication’. This method returns array of object array.

3) Add a method Registration\_data() to your Test class. This method takes two strings as input parameters.

4) Write script for LogIn Application under method @Test.

Test Case will look like this:

package automationFramework;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.annotations.DataProvider;

import org.testng.annotations.Test;

public class DataProviderTest {

    private static WebDriver driver;

  @DataProvider(name = "Authentication")

  public static Object[][] credentials() {

// The number of times data is repeated, test will be executed the same no. of times

// Here it will execute two times

        return new Object[][] { { "testuser\_1", "Test@123" }, { "testuser\_1", "Test@123" }};

  }

  // Here we are calling the Data Provider object with its Name

  @Test(dataProvider = "Authentication")

  public void test(String sUsername, String sPassword) {

      driver = new FirefoxDriver();

      driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

      driver.get("http://www.store.demoqa.com");

      driver.findElement(By.xpath(".//\*[@id='account']/a")).click();

// Argument passed will be used here as String Variable

      driver.findElement(By.id("log")).sendKeys(sUsername);

      driver.findElement(By.id("pwd")).sendKeys(sPassword);

      driver.findElement(By.id("login")).click();

      driver.findElement(By.xpath(".//\*[@id='account\_logout']/a")).click();

      driver.quit();

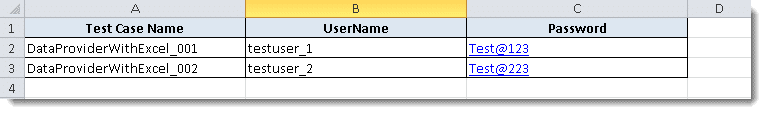
  }

}

## Step 2:  Create a Test Data sheet

1) Create a ‘[**New Package**](http://toolsqa.com/selenium-webdriver/configure-eclipse-with-selenium-webdriver/)‘ file and name it as ‘**testData’**, by right click on the Project and select **New** > **Package**. I always place my Test Data file under separate test data folder.

2) Place a **Excel** file in the above created package location and save it as **TestData.xlsx**. Fill the data in the excel like below image:

[](http://toolsqa.com/wp-content/uploads/2014/05/TestNg-Excel.png)

## Step 3: Create functions to Open & Read data from Excel

We need a way to open this Excel sheet and read data from it within our Selenium test script. For this purpose, I use the Apache POI library, which allows you to read, create and edit Microsoft Office-documents using Java. The classes and methods we are going to use to read data from Excel sheet are located in the org.apache.poi.hssf.usermodel package.

To see step by step process to set up Apache POI Excel, please visit [**Data Driven Framework.**](http://toolsqa.com/selenium-webdriver/data-driven-testing-excel-poi/)

package utility;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import org.apache.poi.xssf.usermodel.XSSFCell;

import org.apache.poi.xssf.usermodel.XSSFRow;

import org.apache.poi.xssf.usermodel.XSSFSheet;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

public class ExcelUtils {

private static XSSFSheet ExcelWSheet;

private static XSSFWorkbook ExcelWBook;

private static XSSFCell Cell;

private static XSSFRow Row;

public static Object[][] getTableArray(String FilePath, String SheetName) throws Exception {

String[][] tabArray = null;

try {

FileInputStream ExcelFile = new FileInputStream(FilePath);

// Access the required test data sheet

ExcelWBook = new XSSFWorkbook(ExcelFile);

ExcelWSheet = ExcelWBook.getSheet(SheetName);

int startRow = 1;

int startCol = 1;

int ci,cj;

int totalRows = ExcelWSheet.getLastRowNum();

// you can write a function as well to get Column count

int totalCols = 2;

tabArray=new String[totalRows][totalCols];

ci=0;

for (int i=startRow;i<=totalRows;i++, ci++) {

cj=0;

for (int j=startCol;j<=totalCols;j++, cj++){

tabArray[ci][cj]=getCellData(i,j);

System.out.println(tabArray[ci][cj]);

}

}

}

catch (FileNotFoundException e){

System.out.println("Could not read the Excel sheet");

e.printStackTrace();

}

catch (IOException e){

System.out.println("Could not read the Excel sheet");

e.printStackTrace();

}

return(tabArray);

}

public static String getCellData(int RowNum, int ColNum) throws Exception {

try{

Cell = ExcelWSheet.getRow(RowNum).getCell(ColNum);

int dataType = Cell.getCellType();

if (dataType == 3) {

return "";

}else{

String CellData = Cell.getStringCellValue();

return CellData;

}catch (Exception e){

System.out.println(e.getMessage());

throw (e);

}

}

}

## Step 4: Create a TestNg test case for accepting data from Excel using Data Provider

1) Create a TestNG class ‘DataProviderWithExcel’ by Pressing Ctrl+N , select ‘**Create TestNG Class**‘ under TestNG category and Under Annotations, check ‘**@BeforeMethod**‘, ‘**@AfterMethod**‘ & ‘**DataProvider**‘ and click Finish.

3) Add a method Registration\_data() to your Test class. This method takes two strings as input parameters.

4) Now divide the test case in to three parts :

@BeforeMethod : Launch Firefox and direct it to the Base URL

@Test : Enter Username & Password to Login, Print console message and Log out

@AfterMethod : Close Firefox browser

Test Case will look like this:

package practiceTestCases;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.Test;

import org.testng.annotations.DataProvider;

import utility.ExcelUtils;

public class DataProviderWithExcel\_001 {

WebDriver driver;

@BeforeMethod

public void beforeMethod() throws Exception {

driver = new FirefoxDriver();

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

driver.get("http://www.store.demoqa.com");

}

@Test(dataProvider="Authentication")

public void Registration\_data(String sUserName,String sPassword)throws Exception{

driver.findElement(By.xpath(".//\*[@id='account']/a")).click();

driver.findElement(By.id("log")).sendKeys(sUserName);

System.out.println(sUserName);

driver.findElement(By.id("pwd")).sendKeys(sPassword);

System.out.println(sPassword);

driver.findElement(By.id("login")).click();

System.out.println(" Login Successfully, now it is the time to Log Off buddy.");

driver.findElement(By.xpath(".//\*[@id='account\_logout']/a")).click();

}

@DataProvider

public Object[][] Authentication() throws Exception{

Object[][] testObjArray = ExcelUtils.getTableArray("D://ToolsQA//OnlineStore//src//testData//TestData.xlsx","Sheet1");

return (testObjArray);

}

@AfterMethod

public void afterMethod() {

driver.close();

}

}

## Step 5: Run the test against the Test Case name in the Test Data file

1) This means that your test should be run once only with the data which is mentioned against the Test Case name. To do this we need to tweak the Excel utility class, plus need to add few more function to fetch out the current Test Case name & the row number which contain the Test Case name.

package utility;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import org.apache.poi.xssf.usermodel.XSSFCell;

import org.apache.poi.xssf.usermodel.XSSFRow;

import org.apache.poi.xssf.usermodel.XSSFSheet;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

public class ExcelUtils {

private static XSSFSheet ExcelWSheet;

private static XSSFWorkbook ExcelWBook;

private static XSSFCell Cell;

private static XSSFRow Row;

//This method is to set the File path and to open the Excel file, Pass Excel Path and Sheetname as Arguments to this method

public static void setExcelFile(String Path,String SheetName) throws Exception {

try {

// Open the Excel file

FileInputStream ExcelFile = new FileInputStream(Path);

// Access the required test data sheet

ExcelWBook = new XSSFWorkbook(ExcelFile);

ExcelWSheet = ExcelWBook.getSheet(SheetName);

} catch (Exception e){

throw (e);

}

}

public static Object[][] getTableArray(String FilePath, String SheetName, int iTestCaseRow) throws Exception

{

String[][] tabArray = null;

try{

FileInputStream ExcelFile = new FileInputStream(FilePath);

// Access the required test data sheet

ExcelWBook = new XSSFWorkbook(ExcelFile);

ExcelWSheet = ExcelWBook.getSheet(SheetName);

int startCol = 1;

int ci=0,cj=0;

int totalRows = 1;

int totalCols = 2;

tabArray=new String[totalRows][totalCols];

for (int j=startCol;j<=totalCols;j++, cj++)

{

tabArray[ci][cj]=getCellData(iTestCaseRow,j);

System.out.println(tabArray[ci][cj]);

}

}

catch (FileNotFoundException e)

{

System.out.println("Could not read the Excel sheet");

e.printStackTrace();

}

catch (IOException e)

{

System.out.println("Could not read the Excel sheet");

e.printStackTrace();

}

return(tabArray);

}

//This method is to read the test data from the Excel cell, in this we are passing parameters as Row num and Col num

public static String getCellData(int RowNum, int ColNum) throws Exception{

try{

Cell = ExcelWSheet.getRow(RowNum).getCell(ColNum);

String CellData = Cell.getStringCellValue();

return CellData;

}catch (Exception e){

return"";

}

}

public static String getTestCaseName(String sTestCase)throws Exception{

String value = sTestCase;

try{

int posi = value.indexOf("@");

value = value.substring(0, posi);

posi = value.lastIndexOf(".");

value = value.substring(posi + 1);

return value;

}catch (Exception e){

throw (e);

}

}

public static int getRowContains(String sTestCaseName, int colNum) throws Exception{

int i;

try {

int rowCount = ExcelUtils.getRowUsed();

for ( i=0 ; i<rowCount; i++){

if (ExcelUtils.getCellData(i,colNum).equalsIgnoreCase(sTestCaseName)){

break;

}

}

return i;

}catch (Exception e){

throw(e);

}

}

public static int getRowUsed() throws Exception {

try{

int RowCount = ExcelWSheet.getLastRowNum();

return RowCount;

}catch (Exception e){

System.out.println(e.getMessage());

throw (e);

}

}

}

**Final Test Case**

1) Get the Test Case name.

2) With the Test Case name, get the row number of the Excel sheet for the test.

3) Get the data from the excel sheet for the fetched test row.

package practiceTestCases;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.annotations.Test;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.DataProvider;

import utility.ExcelUtils;

public class DataProviderWithExcel\_002 {

private String sTestCaseName;

private int iTestCaseRow;

WebDriver driver;

@BeforeMethod

public void beforeMethod() throws Exception {

driver = new FirefoxDriver();

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

driver.get("http://www.store.demoqa.com");

}

@Test(dataProvider = "Authentication")

public void f(String sUserName, String sPassword) {

driver.findElement(By.xpath(".//\*[@id='account']/a")).click();

driver.findElement(By.id("log")).sendKeys(sUserName);

System.out.println(sUserName);

driver.findElement(By.id("pwd")).sendKeys(sPassword);

System.out.println(sPassword);

driver.findElement(By.id("login")).click();

System.out.println(" Login Successfully, now it is the time to Log Off buddy.");

driver.findElement(By.xpath(".//\*[@id='account\_logout']/a")).click();

}

@AfterMethod

public void afterMethod() {

driver.close();

}

@DataProvider

public Object[][] Authentication() throws Exception{

// Setting up the Test Data Excel file

ExcelUtils.setExcelFile("D://ToolsQA//OnlineStore//src//testData//TestData.xlsx","Sheet1");

sTestCaseName = this.toString();

// From above method we get long test case name including package and class name etc.

// The below method will refine your test case name, exactly the name use have used

sTestCaseName = ExcelUtils.getTestCaseName(this.toString());

// Fetching the Test Case row number from the Test Data Sheet

// Getting the Test Case name to get the TestCase row from the Test Data Excel sheet

iTestCaseRow = ExcelUtils.getRowContains(sTestCaseName,0);

Object[][] testObjArray = ExcelUtils.getTableArray("D://ToolsQA//OnlineStore//src//testData//TestData.xlsx","Sheet1",iTestCaseRow);

return (testObjArray);

}

}