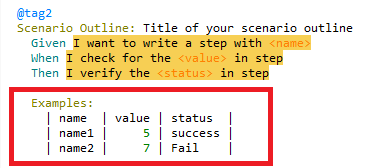
**¿Cómo Usar una fuente de Datos Externa para nuestros Script (Excel)?**

Hasta el momento hemos usando nuestros datos desde la historia de usuario (.Feature) para las Automatizaciones que estamos realizando.



Sin embargo, en ocasiones necesitamos utilizar un conjunto amplio de datos y que mejor que Excel para poder hacer esto.

**Objetivo:**

Realizar la ejecución de nuestros escenarios con datos tomados de un libro de Excel.

**Pasos para configurar nuestro proyecto**

1. Debemos adicionar las siguientes dependencias a nuestro archivo **POM.XML**

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi</artifactId>

<version>3.17</version>

</dependency>

<dependency>

<groupId>org.apache.poi</groupId>

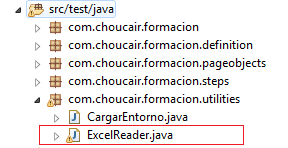
<artifactId>poi-ooxml</artifactId>

<version>3.17</version>

</dependency>

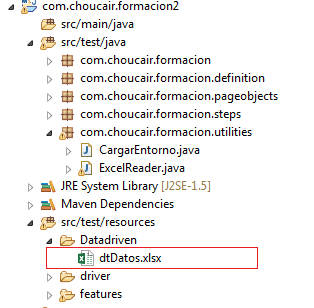
**Nota: Recuerda que debes estar en una red con navegación abierta para que descargue estos .jar**

1. Agregar la clase ExcelReader.java al paquete de utilidades de nuestro proyecto.



|  |
| --- |
| import java.io.File;  import java.io.FileInputStream;  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;  import org.apache.poi.ss.usermodel.Cell;  public class ExcelReader {  private static XSSFSheet ExcelWSheet;  private static XSSFWorkbook ExcelWBook;  private static XSSFCell Cell;  private static XSSFRow Row;  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);  }    }  @SuppressWarnings("deprecation")  public static String getCellData(int RowNum, int ColNum) throws Exception{  String CellData = "";  try{  Cell = ExcelWSheet.getRow(RowNum).getCell(ColNum);  /\*Consultar el tipo de Celda: (0 = numérico, 1 = String)  Cell.getCellType(); \*/  if (Cell.getCellType()==1){ CellData = Cell.getStringCellValue(); }  if (Cell.getCellType()==0){ CellData = Cell.getRawValue(); }  return CellData;  }catch (Exception e){  return "";  }  }  public static int ContarFilas() throws Exception  {  int NumFilas = ExcelWSheet.getLastRowNum();  return NumFilas;  }  public static void setCellData(int RowNum, int ColNum, String TextObtenido) {  try {  ExcelWSheet.getRow(RowNum).createCell(ColNum).setCellValue(TextObtenido);  ExcelWSheet.getRow(RowNum).getCell(ColNum).setCellValue(TextObtenido);  }  catch (Exception e)  {  System.out.println(e.getMessage());  }  }  public static void SaveData(String Path) throws IOException {  try {  FileOutputStream ExcelFileOut = new FileOutputStream(Path);  ExcelWBook.write(ExcelFileOut);  }  catch (Exception e)  { System.out.println(e.getMessage()); }  }  public static void CerrarBook() throws IOException {  ExcelWBook.close();  }  } |

1. En el folder de “resources” de nuestro proyecto, crear un folder “Datadriven” y adjuntar el archivo de Excel a leer.

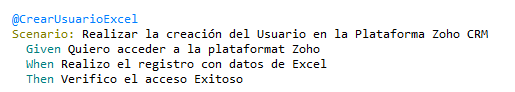


Ejemplos de implementación:

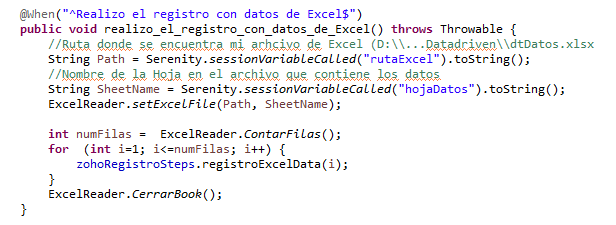
|  |  |
| --- | --- |
| //setExcelFile: Abrir un archivo de Excel  //Parámetros:  //Parm1 = Ruta y nombre del archivo Excel a cargar  //Parm2 = Nombre de la hoja que contiene los datos  ExcelReader.*setExcelFile*(FileExcel, "Hoja1"); | //getCellData : devuelve el contenido de una celda  //Parámetros:  //Parm1 = Fila  //Parm2 = Columna  **Stgring** Datos **=** ExcelReader.*getCellData*(i, 1) |
| //ContarFilas: devuelve el nro de registros existentes en el archivo  //Parámetros: Sin parámetros  **int** numFilas = ExcelReader.*ContarFilas*();  **for** (**int** i=1; i<=numFilas; i++) { | //setCellData : devuelve el contenido de una celda  //Parámetros:  //Parm1 = Fila  //Parm2 = Columna  //Parm3 = Valor a almacenar  ExcelReader.*setCellData*(i, 6, "Valor almacenar") |
| // SaveData: Guardar cambios sin cerrar  //Parámetros:  //Parm1 = Nombre del archivo a guardar, para copias asigne un nuevo nombre  ExcelReader.*SaveData*(FileExcel); | // CerrarBook: Cerrar el libro  //Parámetros: NA  ExcelReader.*CerrarBook*(); |

**¿Cómo implemento esta solución en mi proyecto?**

**Feature**



**Definition**



**Step**



**PageObject**

