Apache POI

Testleaf
Always Ahead

Summary

- Java Library to read or write the data to MS Office application like PPT, Excel, word etc.,
- Supports dynamic data parameterization in selenium (data driven testing)

Pre-requisite

-to integrate the data files with Selenium, dependencies should be added to the pom.xml

```
<dependency>
   <groupId>org.apache.poi</groupId>
   <artifactId>poi</artifactId>
   <version>5.1.0</version>
</dependency>
<dependency>
   <groupId>org.apache.poi</groupId>
   <artifactId>poi-ooxml</artifactId>
   <version>5.1.0</version>
</dependency>
```

poi : to integrate excel sheet with .xls format(Excel 97-2003 workbook)

poi-ooxml: to integrate the excel sheet with .xlsx format(Xml formatted Excel Workbook >2003)

HSSFWorkbook: **It is a high-level class under the org.apache.poi.hssf.usermodel** package. It implements the Workbook interface and is used for Excel files in .xls format.

Components

XSSFWorkbook*:*

Constructs/identify the XSSFWorkbook object given the full path of a file

XSSFWorkbook wb = new XSSFWorkbook ("./data/Excelfile.xlsx)"

XSSFSheet : *class under* the **org.apache.poi.** *xssf***.usermodel** *package. It can create excel spreadsheets and it allows to interact with the sheets*

XSSFSheet sheet =book.getSheetAt(0);

getSheetAt(0);uses sheet index

getSheetName("sheet1"); uses sheet name

XSSFRow: *This is a class under the* ***org.apache.poi.xssf.usermodel*** *package. It implements the Row interface, therefore it helps to access the rows in the Sheet*

XSSFCell: This is a class under **the org.apache.poi.xssf.usermodel package.** It implements the Cell interface. *therefore it helps to access the data in column wise*

Methods with XSSF

getRow(): returns the row with the index starts from 0 from the given sheet(XSSFRow)

getLastRowNum(): *returns the number of active rows in the given sheet(row with data) and excludes the first row in Excel sheet as it considers header by default(integer data)*

getPhysicalNumberOfRows(): returns the defined physical number of rows (includes header)

getCell() : returns the cell with the index starts from 0 (represents the column number) from the given sheet(XSSFCell)

getLastCellNum(): *returns the number of active column in the given sheet(columns with data) (integer data)*

Sample Code

```
//Step:1->set the path for the excelsheet
 XSSFWorkbook book=new XSSFWorkbook("./Exceldata/SampleData.xlsx");

 //Step:2->identify the sheet
 XSSFSheet sheet = book.getSheetAt(0);

 //book-->sheet->read the row-->col-->read the values
 XSSFRow row = sheet.getRow(1);
 XSSFCell cell = row.getCell(2);
 String stringCellValue = cell.getStringCellValue();
 System.out.println(stringCellValue);

 int rowCount= sheet.getLastRowNum();//by default it considers the first row as header
 System.out.println(rowCount);

 int cellCount = row.getLastCellNum();
 System.out.println(cellCount);

 for (int i = 1; i <= rowCount; i++) {
         for (int j = 0; j < cellCount; j++) {
                 //book-->sheet->read the row-->col-->read the values
                 String datas = sheet.getRow(i).getCell(j).getStringCellValue();
                 System.out.print(datas);
         }           }
```