

Selenium Apache POI

- Collection of pure Java libraries & is used to read / write MS Office documents such as Excel, Word, PPT, etc
- Helps to design a cross-platform API that can manipulate various file formats of MS Office
- It consists of components such as POIFS & HSSF to access MS Office documents
- HSSF – used to access MS Excel 97 (xls format)
- XSSF – used to access Excel 2007+ (xlsx format)

Features of POI:

- POI – Poor Obfuscation Implementation File System
- It provides stream-based processing that is useful for large files
- It is helpful to handle both XLS & XLSX formats
- Libraries offered for features such as working with formula, creating cell styles with color & border, font, header, footer, data validation, etc.

Methods:

- ✓ createSheet()- create a new sheet in the existing workbook
- ✓ write() – will write data in the sheet
- ✓ createRow() – to create a new row after an existing row
- ✓ createCell() – to create a new cell in an existing row
- ✓ getSheet() – to fetch an existing sheet
- ✓ getRow() – to fetch an existing row
- ✓ getCell() – to fetch an existing cell

Example 1: To write data in a new excel document

```
package selenium_Week3;
```

```
import java.io.FileNotFoundException;
```

```
import java.io.FileOutputStream;
```

```
import java.io.IOException;
```

```
import java.io.OutputStream;
```

```
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
```

```
import org.apache.poi.ss.usermodel.Cell;
```

```
import org.apache.poi.ss.usermodel.Row;
```

```
import org.apache.poi.ss.usermodel.Sheet;
```

```
import org.apache.poi.ss.usermodel.Workbook;
```

```

public class Ex1_WriteSheet {

    public static void main(String[] args) throws
IOException {

        Workbook wb = new HSSFWorkbook();
        OutputStream fileOut = new
FileOutputStream("sample.xls");

        Sheet sheet1 = wb.createSheet("First");
        Sheet sheet2 = wb.createSheet("Second");

        Row row = sheet1.createRow(2);
        Cell cell = row.createCell(5);
        cell.setCellValue("John");

        wb.write(fileOut);

    }

}

```

Example 2: To read a particular data from excel

```

package selenium_Week3;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;

import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.Cell;
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;

```

```

public class Ex2_ReadExcel {

    public static void main(String[] args) throws
    EncryptedDocumentException, IOException {

        InputStream fileIn = new
        FileInputStream("E:\\Selenium\\Programs\\CSDQEA24SD
        1234_Selenium\\Data\\InputData.xls");
        Workbook wb =
        WorkbookFactory.create(fileIn);

        Sheet sheet = wb.getSheetAt(0);
        Row row = sheet.getRow(0);
        Cell cell = row.getCell(1);

        if (cell != null)
            System.out.println("Data in excel : " +
cell);
        else
            System.out.println("Cell is empty");
    }

}

```

Example 3: To read data from all rows & columns is row size & column size not known

```

package selenium_Week3;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.util.Iterator;

import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.Cell;

```

```

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;

public class Ex3_ReadExcel {

    public static void main(String[] args) throws
    EncryptedDocumentException, IOException {

        InputStream fileIn = new
        FileInputStream("E:\\Selenium\\Programs\\CSDQEA24SD
        1234_Selenium\\Data\\InputData.xls");
        Workbook wb =
        WorkbookFactory.create(fileIn);

        Sheet sheet = wb.getSheet("Sheet1");
        Iterator<Row> rowIterator =
        sheet.iterator();

        while(rowIterator.hasNext()){
            Row row1 = rowIterator.next();
            Iterator<Cell> cellIterator =
            row1.iterator();

            while(cellIterator.hasNext()){
                Cell cell1 = cellIterator.next();

                System.out.print(cell1.getStringCellValue() +
                "--");
            }
            System.out.println();
        }
        wb.close();
    }
}

```

