

Excel Support

Service suite offers excellent Excel support from the lead open source POI project

Basic calls are

`readExcel(Path:String) → ExcelWrapper`

Read an excel and returns the Excel Wrapper object, for Excel Wrapper methods, check the Excel Wrapper section above. Keep in mind you can use “{ds}” to indicate path relative to the current script.

`writeExcel(Data: ExcelWrapper, Path:String) → ExcelWrapper`

Writes an excel wrapper to disk, keep in mind you can use “{ds}” to indicate path relative to the current script. This functionality works in IDE but not during scenario execution.

`createNewExcel(FirstSheetName:String) → ExcelWrapper`

Creates a new excel wrapper with one sheet. For Excel Wrapper methods, check the Excel Wrapper section above.

ExcelWrapper

Excel wrapper holds a POI 5.0.0 document. It has basic cell editing, for full POI documentation

<https://poi.apache.org/>

`setValueAt(row:int, column:int, value:Object)`

Sets a value in a specific cell

```
var sheet = readExcel("myExcel.xlsx");
sheet.setValueAt(2,1,"P0000002643");
```

getValueAt(row:int, column:int) → value:String

Gets the value from a specific cell in String format. If Cell is of type NUMERIC an exception will be thrown.

```
var sheet = readExcel("myExcel.xlsx");
log("Value: "+sheet.getValueAt(2,1));
```

For getting the String representation of a cell of type numeric or date can use **toString** method of [XSSFCell](#) object returned by **getCellAt** (see below).

```
var cell = sheet.getCellAt(2,5);
log("cell string representation: "+cell.toString());
```

getCellAt(row:int, column:int) → XSSFCell

Gets the XSSFCell object. More info can be found in [XSSFCell](#)

```
var sheet = readExcel("myExcel.xlsx");
var dateCell = sheet.getCellAt(2,5);
log("Date: "+dateCell.toString());
```

```
// COPYING A CELL DATE
var excel = readExcel("myExcel.xlsx");
var cell25 = excel.getCellAt(2, 5); // value is Date 01-Jan-2020
log("2,5 date value: " + cell25.toString());
cell27 = excel.getCellAt(2, 7)
log("Cell 2,7 raw value: " + cell27.getRawValue()); // null

var policy = org.apache.poi.ss.usermodel.CellCopyPolicy();
cell27.copyCellFrom(cell25,policy);
log("Cell 2,7 value after copying cell 2,5: " + cell27.toString()); // 01-Jan-2020
```

getWorkbook() → XSSFWorkbook

Gets the workbook, full documentation about workbook can be found

<https://poi.apache.org/apidocs/dev/org/apache/poi/xssf/usermodel/XSSFWorkbook.html>

setBooleanValueAt(row:int, column:int, value:boolean) → void

Set a cell with a boolean value

```
var sheet = readExcel("myExcel.xlsx");
sheet.setBooleanValueAt(2,3,true);
assertEquals(true, sheet.getBooleanValueAt(2,3))
```

getBooleanValueAt(row:int, column:int) → value:boolean

Get boolean value of a cell

```
var sheet = readExcel("myExcel.xlsx");
sheet.setBooleanValueAt(2,3,true);
assertEquals(true, sheet.getBooleanValueAt(2,3))
```

setFormulaValueAt(row:int, column:int, formula:string) → void

Set a cell's formula

```
var sheet = readExcel("myExcel.xlsx");
sheet.setFormulaValueAt(3,3, 'AVERAGE("E1:G1")');
assertEquals('AVERAGE("E1:G1")', sheet.getFormulaAt(3,3))
```

getFormulaAt(row:int, column:int) → formula:string

Get a cell's formula as a string

```
var sheet = readExcel("myExcel.xlsx");
sheet.setFormulaValueAt(3,3, 'AVERAGE("E1:G1")');
assertEquals('AVERAGE("E1:G1")', sheet.getFormulaAt(3,3))
```

setCellType(row:int, column:int, formula:string, CellType value) → void

Set a cell's cell type as in [Cell Type](#)

```
var sheet = readExcel("myExcel.xlsx");
var cell = sheet.getCellAt(2,4);
cell.setCellType(org.apache.poi.ss.usermodel.CellType.BLANK);
```

getCellType(row:int, column:int) → formula:[CellType](#)

Get a cell's Cell Type

```
var sheet = readExcel("myExcel.xlsx");
assertEquals('NUMERIC', sheet.getCellType(2,3).toString());
```

setDateValueAt(row:int, column:int, value:[Date](#)) → void

Set a cell's cell type as in [Cell Type](#)

```
var sheet = readExcel("myExcel.xlsx");
var cell = sheet.setCellDateAt(2,4, (new java.util.Date).now());
```

getDateValueAt(row:int, column:int) → date:[Date](#)

Get a cell's Date object

```
var sheet = readExcel("myExcel.xlsx");
log('NUMERIC', sheet.getDateValueAt(2,3));
```

setNumericValueAt(row:int, column:int, formula:string, value:double) → void

Set a cell's cell numeric value

```
var sheet = readExcel("myExcel.xlsx");  
var cell = sheet.setNumericValueAt(2,5, 43831);
```

getNumericValueAt(row:int, column:int) → value:double

Get a cell's numeric value as a string

```
var sheet = readExcel("myExcel.xlsx");  
log(43831.0, sheet.getDateValueAt(2,3));
```

getRawValueAt(row:int, column:int) → value:string

Get a cell's raw value formatted as a string

```
var sheet = readExcel("myExcel.xlsx");  
assertEquals('43831.0', sheet.getRawValueAt(2,3));
```

getFormattedValueAt(row:int, column:int) → value:string

Get a cell's formatted value as a string

```
var sheet = readExcel("myExcel.xlsx");  
assertEquals('1/1/20', sheet.getFormattedValueAt(2,3));
```

setDateStyleAt(row:int, column:int, format:string) → void

Sets a cell's format style

```
var workbook = createNewExcel("Hello World");  
workbook.setNumericValueAt(2,2,43831); // Jan/01/2020  
workbook.setDateStyleAt(2,2,"mm/dd/yy");  
assertEquals("01/01/20", workbook.getFormattedValueAt(2,2));
```

Revision #1

Created 2026-07-24 14:33:33 UTC by Admin

Updated 2026-07-24 14:33:33 UTC by Admin