Aspose Cells Functionalities

# **Overview**

Aspose.Cells for Java is a programming class library that allows software developers to manipulate and process spreadsheet files within their own applications. Aspose.Cells for Java supports the popular spreadsheet (XLS, XLSX, XLSM, XLSB, XLTX, XLTM, CSV, SpreadsheetML, ODS) file formats your business use every day. It also allows exporting Excel files to PDF, DOCX, PPTX, JSON, XPS, HTML, MHTML, Plain Text and popular image formats including TIFF, JPG, PNG, BMP and SVG.
To purchase Aspose Cells Java license, kindly refer to the link. [Buy now - Purchase - Aspose](https://purchase.aspose.com/buy)

# **Functions**

* Exportexcelsheetdata()

# **Smart Services**

* Aspose Convert Excel
* Aspose Copy Sheet Between Workbooks
* Aspose Copy Sheet Within Workbook
* Aspose Decrypt Excel
* Aspose Encrypt Excel
* Aspose Import Data into Excel (JSON)
* Aspose Insert Image into Excel
* Aspose Merge Workbook
* Aspose Merge Worksheet
* Aspose Move Sheet Within Workbook

# **Conversion**

## **Parameters**

### **Inputs**

* sourceDocument(Document) - The source document can be in any of the following file formats: xls, xlsx, xlsm, xlsb, xltx, xlt, xltm, ods, csv, tsv, json.
* newDocumentName (Text) - Name for the newly created document.
* createdDocumentDescription (Text) - Description of the output document.
* convertAs(Text) - The document can be converted to the following file formats: pdf, xls, xlsx, docx, pptx, xlsm, xlsb, xlt, xltx, xltm, ods, ots, csv, tsv, jpg, bmp, png, webp, svg, tiff, xps, md, json, xml, zip,txt
* saveInFolder (Folder) - Folder in which the output document should be saved.
* asposeLicense(Document) - The license document from Aspose.Cells

### **Outputs**

* createdDocument (List of Documents) - The output document file.
* errorMessage (Text) - Error message received.
* errorOccurred(Boolean) - Set to true on the occurrence of error

### **Screenshot**



# **Merge Workbook**

## **Parameters**

### **Inputs**

* sourceDocuments(List of Documents) The source document can be of the following file formats XLS, XLSX, XLSM, XLSB, ODS, CSV, TSV, HTML, HTM, MHT, MHTML, NUMBERS
* newDocumentName (Text) - Name for the newly created document.
* createdDocumentDescription (Text) - Description of the output document.
* saveAs(Text) - The document can be saved as the following file formats PDF, XLS, XLSX, DOCX, XLSM, XLSB, ODS, CSV, TSV, HTML, XPS, MHTML, MD
* saveInFolder (Folder) - Folder to which the output document is to be saved in.
* asposeLicense(Document) - The license document from Aspose.Cells

### **Outputs**

* createdDocument (Document) - The output Document.
* errorMessage(Text) - Error message received.
* errorOccurred(Boolean) - Set to true on the occurrence of error

### **Screenshot**



# **Merge Worksheet**

## **Parameters**

### **Inputs**

* sourceDocument(Document) -The source document can be of the following file formats XLS, XLSX, XLSM, XLSB, ODS, CSV, TSV, HTML, HTM, MHT, MHTML, NUMBERS
* newDocumentName (Text) - Name for the newly created document.
* createdDocumentDescription (Text) - Description of the output document.
* saveAs(Text) - The document can be saved as the following file formats PDF, XLS, XLSX, DOCX, XLSM, XLSB, ODS, CSV, TSV, HTML, XPS, MHTML, MD
* saveInFolder (Folder) - Folder to which the output document is to be saved in.
* asposeLicense(Document) - The license document from Aspose.Cells

### **Outputs**

* createdDocument (Document) - The output Document.
* errorMessage(Text) - Error message received.
* errorOccurred(Boolean) - Set to true on the occurrence of error

### **Screenshot**



# **Copy Sheet Between Workbooks**

## **Parameters**

### **Inputs**

* fromDocument(Document) The source document from which the sheet should be copied, Valid file formats are XLS, XLSX.
* toDocument(Document) - The target document to which the sheet should be copied, Valid file formats are XLS, XLSX.
* fromSheetNumber(Number) - The sheet number of the source document to be copied. The index starts with 0. 0 represents the first sheet.
* toSheetNumber(Number) - The sheet number of the to document to be pasted. The index starts with 0. 0 represents the first sheet.
* newDocumentName (Text) - Name for the newly created document.
* createdDocumentDescription (Text) - Description of the output document.
* saveAs(Text) - The document can be saved as the following file formats XLS, XLSX.
* saveInFolder (Folder) - Folder to which the output document is to be saved in.
* asposeLicense(Document) - The license document from Aspose.Cells

### **Outputs**

* createdDocument (Document) - The output Document.
* errorMessage(Text) - Error message received.
* errorOccurred(Boolean) - Set to true on the occurrence of error

### **Screenshot**



# **Copy Sheet Within Workbook**

## **Parameters**

### **Inputs**

* fromDocument(Document) The source document from which the sheet should be copied, Valid file formats are XLS, XLSX.
* sheetNumber(Number) - The sheet number of the source document to be copied. The index starts with 0. 0 represents the first sheet.
* newDocumentName (Text) - Name for the newly created document.
* createdDocumentDescription (Text) - Description of the output document.
* saveAs(Text) - The document can be saved as the following file formats XLS, XLSX.
* saveInFolder (Folder) - Folder to which the output document is to be saved in.
* asposeLicense(Document) - The license document from Aspose.Cells

### **Outputs**

* createdDocument (Document) - The output Document.
* errorMessage(Text) - Error message received.
* errorOccurred(Boolean) - Set to true on the occurrence of error

### **Screenshot:**



# **Move Sheet Within Workbook**

## **Parameters**

### **Inputs**

* sourceDocument(Document) The source document from which the sheet should be copied, Valid file formats are XLS, XLSX.
* fromSheetNumber(Number) - The sheet number of the source document to be moved. The index starts with 0. 0 represents the first sheet.
* toSheetNumber(Number) - The sheet number of the document where to be pasted. The index starts with 0. 0 represents the first sheet.
* newDocumentName (Text) - Name for the newly created document.
* createdDocumentDescription (Text) - Description of the output document.
* saveAs(Text) - The document can be saved as the following file formats XLS, XLSX.
* saveInFolder (Folder) - Folder to which the output document is to be saved in.
* asposeLicense(Document) - The license document from Aspose.Cells

### **Outputs**

* createdDocument (Document) - The output Document.
* errorMessage(Text) - Error message received.
* errorOccurred(Boolean) - Set to true on the occurrence of error

### **Screenshot**

#

# **Encryption**

## **Parameters**

### **Inputs**

* sourceDocument(Document) - The source document to be encrypted. The file format should be xlsx or xls.
* newDocumentName (Text) - Name for the newly created document.
* createdDocumentDescription (Text) - Description of the output document.
* password (Text) - Password string used to protect the Excel file.
* saveInFolder (Folder) - Folder to which the output document to be saved in.
* asposeLicense(Document) - The license document from Aspose.Cells.

### **Outputs**

* createdDocument (Document) - The output PDF file.
* errorMessage(Text) - Error message received.
* errorOccurred(Boolean) - Set to true on the occurrence of error

### **Screenshot**



# **Decryption**

## **Parameters**

### **Inputs**

* sourceDocument(Document) - The source document to be decrypted. The file format should be xlsx or xls.
* newDocumentName (Text) - Name for the newly created document.
* createdDocumentDescription (Text) - Description of the output document.
* password (Text) - Password string used to unprotect the Excel file.
* saveInFolder (Folder) - Folder to which the output document is to be saved in.
* asposeLicense(Document) - The license document from Aspose.Cells.

### **Outputs**

* createdDocument (Document) - The output PDF file.
* errorMessage(Text) - Error message received.
* errorOccurred(Boolean) - Set to true on the occurrence of error

### **Screenshot**



# **Insert Image into Excel**

## **Parameters**

### **Inputs**

* SourceDocument (Document) - The source Excel document to which the image should be inserted.
* image - Image to be inserted into the Excel document
* Sheet Number (Number) - The sheet number in which the image is inserted, where 0 is the first sheet, 1 is the second sheet, etc.
* Cell Name (Text) - Name of the cell where the image is inserted. Example A1, B5, etc
* newDocumentName (Text) - Name for the created output Excel document.
* createdDocumentDescription (Text) - The description of the created document.
* Save in Folder (Folder) - Folder to which the output Excel document is to be saved.
* asposeLicense (Document) - The license document from Aspose.Cells.

### **Outputs**

* createdDocument (Document) - The output PDF file.
* errorMessage(Text) - Error message received.
* errorOccurred(Boolean) - Set to true on the occurrence of error

**Screenshot**



# **Import Data into Excel (JSON)**

## **Parameters**

**Notes:**

1. Pass either jsonDocument or jsonString in the input parameter.
2. If excelDocument is passed null new Workbook will be created else will update in given one.

### **Inputs**

* jsonDocument(Document) - The JSON document that contains data to be imported into excel.
* jsonString(Text) - The JSON string that contains data to be imported into excel.
* excelDocument (Document) - The source Excel document to which the data should be imported.
* newDocumentName (Text) - Name for the newly created document.
* createdDocumentDescription (Text) - Description of the output document.
* password (Text) - Password string used to protect the Excel file.
* saveInFolder (Folder) - Folder to which the output document is to be saved in.
* asposeLicense(Document) - The license document from Aspose.Cells.

### **Sample Json**

{

 "sheet1": [

 {

 "id": 1,

 "name": "John Doe",

 "department": "Engineering",

 "salary": 75000,

 "contact": {

 "email": "john.doe@example.com",

 "phone": "+1234567890"

 },

 "projects": [

 {

 "name": "Project A",

 "description": "Developing new feature set",

 "status": "In Progress"

 },

 {

 "name": "Project B",

 "description": "Bug fixing and optimization",

 "status": "Completed"

 }

 ]

 },

 {

 "id": 2,

 "name": "Jane Smith",

 "department": "Marketing",

 "salary": 65000,

 "contact": {

 "email": "jane.smith@example.com",

 "phone": "+1987654321"

 },

 "projects": [

 {

 "name": "Campaign X",

 "description": "Launching new product campaign",

 "status": "Planned"

 }

 ]

 }

 ],

 "company\_info": {

 "name": "TechCorp",

 "address": "123 Main Street, Cityville",

 "phone": "+9876543210",

 "founded": "2000-01-01"

 }

}

### **Outputs**

* createdDocument (Document) - The output PDF file.
* errorMessage(Text) - Error message received.
* errorOccurred(Boolean) - Set to true on the occurrence of error

### **Screenshot**





# **Export Excel Sheet Data**

## **Parameters**

### **Inputs**

* excelDocument(Document) - The Excel document from which data should be exported.Valid formats are xlsx, xls.
* licenseDocument (Document) - The license document from Aspose.Cells.
* sheetNumber(Number) - The sheet number of the source document from the data should be exported. The index starts with 0. 0 represents the first sheet.
* startRow - Provide the starting row index from where the data should be read. Index starts from 0.
* endRow - Provide the ending row index up to which the data should be read. Provide -1 to get all rows in the sheet.
* startColumn - Provide the starting Column index from where the data should be read. The index starts from 0.
* endColumn - Provide the ending Column index up to which the data should be read. Provide -1 to get all column in the sheet.
* IncludeNulls - Provide true to include the empty rows and columns in the sheet when getting the output data. Default: false.

### **Outputs**

* Dictionary
	+ success (Boolean)
	+ sheetName (Text)
	+ totalRows (Number)
	+ data (Dictionary)

### **Screenshot**

