

Libraries comparison

	Aspose.Cells	Apache POI	EasyXLS
Support formats	<ul style="list-style-type: none"> • Microsoft Excel: XLS, XLSX, XLSB, XLT, XLTX, XLTM, XLSM, XML • OpenOffice: ODS • Text: CSV, TSV • Web: HTML, MHTML • Numbers: Apple's iWork office suite Numbers app documents 	<ul style="list-style-type: none"> • Microsoft Excel: XLS, XLSX • Document: DOC, DOCX • Powerpoint: PPT, PPTX 	<ul style="list-style-type: none"> • Microsoft Excel: XLS, XLSX, XLSM, XLSB, XML, • Text: CSV, TXT • Web: HTML
Features	<ul style="list-style-type: none"> • Read, write and modify excel spreadsheets. • Excel 2007/2010/2013 from different sources. • Support for additional excel features: insert data, retrieve data from cells based on their datatype, detect Merged Cells, Find Value in Cells • Read CSV File with Multiple Encodings 	<ul style="list-style-type: none"> • Read, write and modify excel spreadsheets. • Support for additional excel features: working with Formulas, creating cell styles. • Extract property sets of the MS-Office files. • Read and write docx extension files of MS-Word. • Read, create, and edit PowerPoint presentations. • Read and write MS-Publisher files. 	<ul style="list-style-type: none"> • Convert some files (HTML, XML, CSV...) to excel files • Read/Write or edit Excel file • Import, export, and format MS Excel

Advantages	<ul style="list-style-type: none"> • Support processing many files 	<ul style="list-style-type: none"> • Open source (free to use) • Provide many interfaces. • Can work with very large spreadsheets but required less memory. 	Support multi programing languages (C#, C++, PHP, J#...)
Disadvantages	<ul style="list-style-type: none"> • Only free for trial version • Unpopular • Limit number of opened files • With free version, the output worksheet has evaluation watermark 	<ul style="list-style-type: none"> • A more complex library than other libraries • Must use specified interfaces for different format files. • Compatibility is poor 	<ul style="list-style-type: none"> • Must purchase to use all features • Unpopular • must use specified interfaces for different format files.
Can open file with protected password	<p>Yes</p> <p>(use setPassword())</p> <p>My team has run code successfully and reported in detail in Trello Sprint 0 – Item 3.1</p>	<p>Yes</p> <p>(Can open protected XLS file (org.apache.poi.hssf.record.crypt) And XLSX file (using org.apache.poi.poifs.crypt)</p> <p>Reference: click this)</p>	No
License	<p>Commercial</p> <p>References</p>	<p>Apache</p> <p>References</p>	<p>MIT</p> <p>References</p>

Speed test

	Aspose.Cell
Creating files	<p>Scenario 1: Creating 50 XLS Files Each Workbook contains five worksheets with 150*50 records in each worksheet Results from Scenario 1: 17.21s</p> <p>Create an XLS File The workbook contains a single worksheet with 10,000 * 30 records Scenario 2 Results: 7.51s</p> <p>Scenario 3 - Creating an XLS File The workbook contains five worksheets with 150 * 56 records in each worksheet Scenario 3 Results: 0.42</p>
Editing/Updating existing Excel Files	<p>Scenario 1 - Load a 10 MB XLS Template File The workbook contains 100 worksheets with 377 * 18 records in each. Each worksheet is filled with similar data. the string value of a cell is each worksheet is replaced and the updated file is saved as a new file. Scenario 1 Results 5.23s</p> <p>Scenario 2 - Load a 10 MB XLS Template File The workbook contains 100 worksheets with 377 * 18 records in each</p>

	<p>worksheet. Each worksheet is filled with similar data. The scenario inserts 100 rows into each worksheet, fill them with string values and then the updated file as a new file.</p> <p>Scenario 2 Results 6.12</p> <p>Scenario 3 - Load a 10 MB XLS Template File</p> <p>The workbook contains 100 worksheets with 377 * 18 records in each. Each worksheet is filled with similar data. The code then inserts 10 columns in each worksheet, fills them with string values and saves the updated file.</p> <p>Scenario 3 Results: 7.10</p>
References	Click here

	Apache POI
Large file - Writing to HSSF	<ul style="list-style-type: none"> Disadvantages: only 65536 lines can be processed at most, otherwise an exception will be thrown Advantages: write cache in the process, do not operate the disk, the last one-time write to the disk, fast
Large files - Writing to XSSF	<ul style="list-style-type: none"> Disadvantages: the speed of writing data is very slow, consumes memory, and overflows memory, such as 1 million Advantages: can write a large amount of data, such as 200000
Large files - Writing to SXSSF	<ul style="list-style-type: none"> Advantages: can write a very large amount of data, such as 1 million or

	more, write data fast, occupy less memory.
References (because there is no speed test on the Apache POI Homepage, so I refer to speed test on other site)	Click here

Among 3 libraries above, my team recommend using Apache POI library because:

1. **Apache POI is completely free (whilst Aspose.Cell and EasyXLS limit their features in free versions)**
2. **Apache POI is very popular. Therefore, it will be easy for us to find support from the Internet (EasyXLS and AsposeCells are unpopular and have very few users).**

References:

[Aspose.Cells for Java | Documentation](#)

[Apache POI - the Java API for Microsoft Documents](#)

[Apache POI - Overview \(tutorialspoint.com\)](#)

[EasyXLS™ | .NET Excel component | Java Excel library | API for Excel](#)