Reading and Writing Excel (XLSX) Files in PHP

March 11, 2022 · 7 min read

Nico Dupont
Co-Founder & CPO at Akeneo

Excel is the most used file format to share data.

In this tutorial, we will learn how to read and write XLSX files in PHP with the help of examples.

Last but not least, we'll parse an Excel file of 1 million lines in PHP in a very fast and memory-efficient manner.

The whole code and examples are available in a GitHub repository; it comes packaged as a simple Symfony project and provides a Docker image.

Best PHP Libraries to Parse and Write Excel Files

Several libraries allow to work with Excel files; they come with their pros and cons.

The most famous is `phpoffice/phpspreadsheet`; it has more than 10k stars on GitHub. This library allows reading, creating, and writing spreadsheet documents in PHP. This library was formerly named `phpoffice/phpexcel`, the project has been deprecated in 2017, and `phpspreadsheet` officially replaced `phpexcel`.

The challenger is `box/spout`; it has more than 4k stars on GitHub. This library allows reading and writing spreadsheet files in a fast and scalable way.

There is no debate regarding the support of different file formats: `phpoffice/phpspreadsheet` supports far more formats, especially the oldest ones which are very convenient when you need to parse XLS files created with Excel 2003 97, or 95.

Format Support	`phpoffice/phpspreadsheet`	`box/spout`
Office Open XML (.xlsx) Excel 2007+	✅	✅
SpreadsheetML (.xml) Excel 2003	✅	❌
BIFF 8 (.xls) Excel 97	✅	❌
BIFF 5 (.xls) Excel 95	✅	❌
Open Document Format (.ods)	✅	✅
CSV	✅	✅

When it comes to performance, the big difference is that `phpoffice/phpspreadsheet` loads the whole XML document in memory while `box/spout` streams the Excel document's parsing, allowing it to consume very little memory.

Which PHP Excel Library should you use?

Use case	Best Library Choice
Read or write modern and old Excel formats	`phpoffice/phpspreadsheet`
Read or write XLSX files efficiently or parse large files.	`box/spout`

There are other libraries, but they don't bring anything more or very different.

Both `phpspreadsheet` and `spout` also cover the classic operations of reading and writing CSV files in PHP.

Install Box/Spout library

Let's start by installing `box/spout` library.

```
composer require box/spout
```

It provides a simple code API to read or create spreadsheets. Its main advantage is to manipulate large files in a memory-efficient manner.

Excel (XLSX) File Example

We'll use an XLSX file with a single tab containing tabular data about movies in the following example.

id	title	poster	overview	release_date	genres
299537	Captain Marvel	https://.../mWII.jpg	The story follows Carol Danvers as she becomes one of the universe\u2019s most [...]	1551830400	Action, Adventure, Science Fiction
299536	Avengers: Infinity War	https://.../3VAd.jpg	As the Avengers and their allies have continued to protect the world from threats [...]	1526346000	Action, Adventure, Science Fiction
157336	Interstellar	https://.../BvIx.jpg	Interstellar chronicles the adventures of a group of explorers who make use of a [...]	1415145600	Adventure, Drama, Science Fiction

Read an Excel File (XLSX)

We parse our existing XLSX file:

1. Create the Excel Reader using `ReaderEntityFactory::createXLSXReader()`
2. Open the XSLX file with `$reader->open($path)`
3. Browse each Sheet of the spreadsheet with `$reader->getSheetIterator()`
4. Browse each Row of the Sheet with `$sheet->getRowIterator()`
5. Browse each Cell of the Row with `$row->getCells()`
6. Access each Value of the Cell with `$cell->getValue()`
7. Close the Excel Reader (and the File) with `$reader->close()`

```
use Box\Spout\Reader\Common\Creator\ReaderEntityFactory;

$path = 'data/movies-100.xlsx';
# open the file
$reader = ReaderEntityFactory::createXLSXReader();
$reader->open($path);
# read each cell of each row of each sheet
foreach ($reader->getSheetIterator() as $sheet) {
    foreach ($sheet->getRowIterator() as $row) {
        foreach ($row->getCells() as $cell) {
            var_dump($cell->getValue());
        }
    }
}
$reader->close();
```

We can use this snippet to import Excel rows into a database. In that case, we'd need to insert or update data for each parsed row.

Write an Excel File (XLSX)

We write in an XLSX file:

1. Create the Excel Writer using `WriterEntityFactory::createXLSXWriter()`
2. Open the XSLX file with `$writer->open($path)` that will allow to create the Excel file

3. Iterate over the rows we want to insert
4. Create each Excel row with `WriterEntityFactory::createRowFromArray($row)`
5. Insert each row in Excel with `$writer->addRow($rowFromValues);`
6. Close the Excel Writer (and the File) with `$writer->close()`

```php
use Box\Spout\Writer\Common\Creator\WriterEntityFactory;

$rows = [
    ['id', 'title', 'poster', 'overview', 'release_date', 'genres'],
    [181808, "Star Wars: The Last Jedi",
"https://image.tmdb.org/t/p/w500/kOVEVeg59E0wsnXmF9nrh6OmWII.jpg", "Rey develops her newly discovered abilities with the guidance of Luke Skywalker, who is unsettled by the strength of her powers. Meanwhile, the Resistance prepares to do battle with the First Order.", 1513123200, "Documentary"],
    [383498, "Deadpool 2",
"https://image.tmdb.org/t/p/w500/to0spRl1CMDvyUbOnbb4fTk3VAd.jpg", "Wisecracking mercenary Deadpool battles the evil and powerful Cable and other bad guys to save a boy's life.", 1526346000, "Action, Comedy, Adventure"],
    [157336, "Interstellar",
"https://image.tmdb.org/t/p/w500/gEU2QniE6E77NI6lCU6MxlNBvIx.jpg", "Interstellar chronicles the adventures of a group of explorers who make use of a newly discovered wormhole to surpass the limitations on human space travel and conquer the vast distances involved in an interstellar voyage.",1415145600,"Adventure, Drama, Science Fiction"]
];
$path = 'data/new-file.xlsx';
$writer = WriterEntityFactory::createXLSXWriter();
$writer->openToFile($path);
foreach ($rows as $row) {
    $rowFromValues = WriterEntityFactory::createRowFromArray($row);
    $writer->addRow($rowFromValues);
}
$writer->close();
```

We can use this snippet to export data to Excel. For instance, by reading rows from a database table to write each row in the XLSX file.

Besides Microsoft Excel, Google Sheets is getting a lot of traction from users who need to collaborate near real-time on business data. Google offers API to create, read and update Google Sheets data using PHP.

Read a Large Excel File Efficiently while Using Low Memory

A common issue while manipulating large files is memory consumption and reading or writing efficiency.

The great news is that Spout takes this challenge into account and allows browsing the file content without loading everything in memory.

Let's take an example of an xlsx file containing 1M of lines; this file weights 200MB.

```
-rw-rw-rw- 1 nico nico 205M mars 12 10:10 data/movies-1000000.xlsx
```

Let's load all its content:

```php
// I'm using here the Symfony Console & StopWatch components
$section = 'read_excel_file';
$this->stopwatch->start($section);
$path = 'data/movies-1000000.xlsx';
$reader = ReaderEntityFactory::createXLSXReader();
$reader->open($path);
# read each cell of each row of each sheet
foreach ($reader->getSheetIterator() as $sheet) {
    foreach ($sheet->getRowIterator() as $row) {
        $cells = $row->getCells();
        foreach ($cells as $cell) {
            // we do nothing, but we want to ensure we browse each cell
        }
    }
}
$this->stopwatch->stop($section);
$output->writeln("I read 1.000.000 rows from the Excel File ".$path);
$output->writeln((string) $this->stopwatch->getEvent($section));
```

And here is the result:

```
I read 1.000.000 rows from the Excel File data/movies-1000000.xlsx
```

```
default/read_excel_file: 8.00 MiB - 104666 ms
```

We load our 1M lines using only 8MB of memory (I'm using PHP8).

When streaming the data reading and processing, you can apply rich transformations while keeping the memory usage low and a fast execution time.

Download the Code and Examples

You can find all the code and examples in this GitHub repository.

It's packaged as a simple Symfony project, a set of commands, it also comes with a Docker image.

Tags: php