Comparison

Comparison with Alternative Libraries

Anonymous · 09/11/2026

Comparison with Alternative Libraries

When choosing a spreadsheet library for PHP, it’s important to understand the differences between available options. This guide compares Derafu Spreadsheet with some popular alternatives.

Derafu Spreadsheet vs PhpSpreadsheet

PhpSpreadsheet is the most widely used PHP spreadsheet library and is actually used internally by Derafu Spreadsheet for some format handlers.

Advantages of Derafu Spreadsheet

- **Simplified API**: The API is more streamlined and consistent across formats.
- **Customizable type casting**: Has a custom and flexible type casting system.
- **Format-agnostic code**: Write code once that works with any spreadsheet format.
- **Easier data access**: Direct methods for working with rows and cells.
- **Native handling of JSON/YAML**: Built-in support for these data exchange formats.
- **JSON inside**: Allows you to easily store and retrieve JSON data inside spreadsheets.
- **Modular approach**: Use only the format handlers you need.
- **Less overhead**: Simpler in-memory structure for data representation.

When to use PhpSpreadsheet instead

- **Advanced Excel features**: If you need complex Excel features like conditional formatting, charts, etc.
- **Cell styling**: If you need detailed control over cell appearance.
- **Formula calculation**: If you need to evaluate Excel formulas.
- **Mature ecosystem**: PhpSpreadsheet has been around longer and has more examples and community code.

Derafu Spreadsheet vs league/csv

league/csv is a focused library for working with CSV files. Internally, Derafu Spreadsheet uses it, as default, for CSV format handling.

Advantages of Derafu Spreadsheet

- **Multiple format support**: Work with many formats using the same code.
- **Automatic type casting**: league/csv returns everything as strings.
- **Higher-level abstractions**: Work with the concepts of sheets, rows, and cells.
- **Easy format conversion**: Convert between CSV and other formats seamlessly.

When to use league/csv instead

- **CSV-only projects**: If you only need to work with CSV files.
- **Memory efficiency for large files**: league/csv has specialized streaming capabilities.
- **CSV-specific features**: If you need advanced CSV features like RFC compliance options.
- **Simplicity**: If you want a more focused, single-purpose library.

Code Comparison Examples

Basic Reading Example

Derafu Spreadsheet:

```
use Derafu\Spreadsheet\SpreadsheetLoader;

$loader = new Loader();
$sheet = $loader->loadFromFile('data.xlsx')->getActiveSheet();

foreach ($sheet->getDataRows() as $row) {
    $id = $row[0]; // Already cast to proper type (int).
    $date = $row[1]; // Already a DateTimeImmutable.
    // Process...
}
```

PhpSpreadsheet:

```
use DateTimeImmutable;
use PhpOffice\PhpSpreadsheet\IOFactory;

$spreadsheet = IOFactory::load('data.xlsx');
$sheet = $spreadsheet->getActiveSheet();

foreach ($sheet->getRowIterator(2) as $row) {
    $cellIterator = $row->getCellIterator();
    $rowData = [];
    foreach ($cellIterator as $cell) {
        $rowData[] = $cell->getValue();
    }
    $id = (int)$rowData[0]; // Depending on your app and code, manual casting needed.
```

```
    $date = new DateTimeImmutable($rowData[1]); // Manual conversion.
    // Process...
}
```

league/csv:

```
use DateTimeImmutable;
use League\Csv\Reader;

$reader = Reader::createFromPath('data.csv');
$reader->setHeaderOffset(0);

foreach ($reader->getRecords() as $record) {
    $id = (int)$record['id']; // Depending on your app and code, manual casting
needed.
    $date = new DateTimeImmutable($record['date']); // Manual conversion.
    // Process...
}
```

Format Conversion Example

Derafu Spreadsheet:

```
use Derafu\Spreadsheet\SpreadsheetLoader;
use Derafu\Spreadsheet\SpreadsheetDumper;

$loader = new Loader();
$dumper = new Dumper();

// Load XLSX and save as CSV.
$spreadsheet = $loader->loadFromFile('data.xlsx');
$dumper->dumpToFile($spreadsheet, 'data.csv');

// Load CSV and save as JSON.
$spreadsheet = $loader->loadFromFile('data.csv');
$dumper->dumpToFile($spreadsheet, 'data.json');
```

With alternative libraries:

```
use League\Csv\Reader as CsvReader;
use PhpOffice\PhpSpreadsheet\IOFactory as PhpSpreadsheetIOFactory;
use PhpOffice\PhpSpreadsheet\Writer\Csv as PhpSpreadsheetCsvWriter;

// Load with PhpSpreadsheet, save as CSV.
$spreadsheet = PhpSpreadsheetIOFactory::load('data.xlsx');
$writer = new PhpSpreadsheetCsvWriter($spreadsheet);
$writer->save('data.csv');
```

```
// To convert CSV to JSON, would require:
$csv = CsvReader::createFromPath('data.csv');
$csv->setHeaderOffset(0);
$records = iterator_to_array($csv->getRecords());
file_put_contents('data.json', json_encode($records, JSON_PRETTY_PRINT));
```

Feature Comparison Table

Feature	Derafu Spreadsheet	PhpSpreadsheet	league/csv
Formats Supported	XLSX, XLS, CSV, ODS, JSON, XML, YAML, HTML, PDF	XLSX, XLS, CSV, ODS, HTML, PDF	CSV only
API Consistency	Excellent	Complex	Simple
Memory Efficiency	Can be better with streaming support	Poor for large files	Excellent
Streaming Support	Soon	Yes	Excellent
Customizable Casting	□	□	□
PSR-7 Integration	□	□	□
JSON/YAML Support	□	□	□
Formula Support	□	□	□
Cell Styling	□	□	□

Conclusion

Derafu Spreadsheet is focused on simplicity and ease of use for handling the data inside spreadsheets, not the styling. It excels when you need:

1. **Unified handling** of multiple file formats.
2. **Clean, intuitive API** for spreadsheet operations.
3. **Automatic type handling** to reduce boilerplate code.
4. **Format conversion** capabilities.
5. **Modern PHP architecture** with a focus on developer experience.

Other libraries may be better suited for specific use cases:

- **PhpSpreadsheet**: For complex Excel features and formatting.
- **league/csv**: For CSV-specific operations and streaming large files.

Choose the tool that best matches your specific requirements and constraints.

Last updated on 09/11/2026