

Docs » Data Handling Category » Spreadsheet Project » Basic Usage

Basic Usage

Basic Usage

Anonymous · 09/09/2026

Basic Usage

This guide covers the fundamental operations with Derafu Spreadsheet: loading, manipulating, and saving spreadsheet data.

Loading a Spreadsheet

You can load a spreadsheet from a file or from a string:

```
<?php

use Derafu\Spreadsheet\SpreadsheetLoader;

// Create a loader.
$loader = new Loader();

// From a file (format auto-detected from extension).
$spreadsheet = $loader->loadFromFile('data.xlsx');

// From a string (must specify format).
$csvString = "header1,header2\nvalue1,value2";
$spreadsheet = $loader->loadFromString($csvString, 'csv');
```

Working with Sheets

A spreadsheet contains one or more sheets, which you can access and manipulate:

```
// Get all sheet names.
$sheetNames = $spreadsheet->getSheetNames();

// Get a specific sheet.
$sheet = $spreadsheet->getSheet('Sheet1');

// Create a new sheet with data.
$spreadsheet->createSheet('NewSheet', [
    ['Header1', 'Header2', 'Header3'],
```

```
    ['Value1', 'Value2', 'Value3'],
    ['Value4', 'Value5', 'Value6']
]);

// Set active sheet.
$spreadsheet->setActiveSheet('NewSheet');

// Get active sheet.
$activeSheet = $spreadsheet->getActiveSheet();

// Check if a sheet exists.
if ($spreadsheet->hasSheet('SomeSheet')) {
    // ...
}

// Remove a sheet.
$spreadsheet->removeSheet('SomeSheet');
```

Working with Rows and Cells

Once you have a sheet, you can access and modify its data:

```
// Get all rows.
$rows = $sheet->getRows();

// Get a specific row.
$firstRow = $sheet->getRow(0);

// Add a new row.
$sheet->addRow(['New', 'Row', 'Data']);

// Get a specific cell.
$value = $sheet->getCell(0, 0); // Row 0, Column 0.
// or with named columns (for associative sheets).
$value = $sheet->getCell(0, 'column_name');

// Update a cell.
$sheet->setCell(0, 0, 'New Value');

// Get header row.
$headers = $sheet->getHeaderRow();

// Get data rows (excludes header for indexed sheets).
$rows = $sheet->getDataRows();
```

Data Types

Thanks to the automatic type casting, you can work with native PHP types:

```
// Numbers are already cast to int/float.
$number = $sheet->getCell(0, 0); // e.g., 123 (int).

// Dates are cast to DateTimeImmutable.
$date = $sheet->getCell(0, 1); // e.g., DateTimeImmutable object.
echo $date->format('Y-m-d'); // "2025-03-12".

// Booleans are properly typed.
$bool = $sheet->getCell(0, 2); // e.g., true (bool).

// You can set any type and it will be properly stored.
$sheet->setCell(0, 3, new DateTimeImmutable());
$sheet->setCell(0, 4, ['array', 'values']);
$sheet->setCell(0, 5, ['nested' => ['object' => 'structure']]);
```

Associative vs. Indexed Sheets

Derafu Spreadsheet supports both associative (column names as keys) and indexed (numeric keys) sheets:

```
// Check if a sheet is associative.
if ($sheet->isAssociative()) {
    // Work with associative data.
    $rowData = $sheet->getRow(0);
    echo $rowData['column_name'];
} else {
    // Work with indexed data.
    $rowData = $sheet->getRow(0);
    echo $rowData[0]; // First column.
}

// Convert between formats.
$associativeSheet = $sheet->toAssociative(); // First row becomes header.
$indexedSheet = $sheet->toIndexed(); // Keys become first row.
```

Saving a Spreadsheet

Once you're done modifying the data, you can save it to a file or get it as a string:

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

```
// Create a dumper.
$dumper = new Dumper();

// Save to a file (format detected from extension).
$dumper->dumpToFile($spreadsheet, 'output.xlsx');

// Convert to a different format.
$dumper->dumpToFile($spreadsheet, 'output.csv');

// Get as a string.
$jsonString = $dumper->dumpToString($spreadsheet, 'json');
```

Creating a Spreadsheet from Scratch

You can also create a spreadsheet from scratch:

```
use Derafu\Spreadsheet\Spreadsheet;

// Create an empty spreadsheet.
$spreadsheet = new Spreadsheet();

// Create a sheet with data.
$spreadsheet->createSheet('Sheet1', [
    ['Name', 'Age', 'Email'],
    ['John Doe', 30, 'john@example.com'],
    ['Jane Smith', 25, 'jane@example.com']
]);

// Create from array.
$data = [
    'Users' => [
        ['Name', 'Age', 'Email'],
        ['John Doe', 30, 'john@example.com'],
        ['Jane Smith', 25, 'jane@example.com']
    ],
    'Products' => [
        ['ID', 'Name', 'Price'],
        [1, 'Product A', 29.99],
        [2, 'Product B', 49.99]
    ]
];

$spreadsheet = Spreadsheet::fromArray($data);

// Save it.
$dumper = new Dumper();
$dumper->dumpToFile($spreadsheet, 'new_spreadsheet.xlsx');
```

Last updated on 09/09/2026