CSV

Csv of Derafu Support

Anonymous · 05/11/2026

CSV

Simplified reading and writing of CSV with `league/csv`. The defaults are the ones that spreadsheets in Spanish use: separator `;`, enclosure `"` and escape `\`.

Method	What it does
`Csv::read($file, $separator = ';', $enclosure = '"', $escape = '\\', $encoding = 'UTF-8')`	The rows of a file.
`Csv::load($content, ...)`	The rows of a text (same parameters).
`Csv::write($data, $file, $separator, $enclosure, $escape)`	Writes records to a file.
`Csv::generate($data, $separator, $enclosure, $escape, $encoding = 'UTF-8')`	The CSV text of some records.
`Csv::send($data, $filename, $separator, $enclosure, $escape, $sendHeaders = true)`	Sends the CSV as a download.
`Csv::validate($file, $separator, $enclosure, $escape, $sampleSize = 5)`	Whether a file looks like a CSV.

Reading

`read()` and `load()` give **a list of rows**, each one a list of texts. The first row is the header like any other, so take it off if the file has one:

```
use Derafu\Support\Csv;

$rows = Csv::read('file.csv');             // [['a', 'b'], ['1', '2'], ['3', '4']]
$rows = Csv::load("x,y\n1,2", ',');        // [['x', 'y'], ['1', '2']]

$header = array_shift($rows);
$records = array_map(fn (array $row) => array_combine($header, $row), $rows);
```

`$encoding` is the encoding of what is read (`'ISO-8859-1'` for a file from an old system): the rows come out in UTF-8.

Writing

`write()`, `generate()` and `send()` take **records**: a list of associative arrays. The first line is the header, with the keys of the first record.

```
$data = [['a' => 1, 'b' => 2], ['a' => 3, 'b' => 4]];

Csv::generate($data);                       // "a;b\n1;2\n3;4\n"
Csv::write($data, 'export.csv');            // The same, in a file.
Csv::send($data, 'export.csv');             // As a download.

Csv::generate($data, encoding: 'ISO-8859-1');  // For the system that wants it.
```

A value with the separator inside is put in the enclosure (`a;b` is written as `"a;b"`). `generate()` can write in another encoding; `send()` always sends UTF-8 (`Content-Type: text/csv; charset=UTF-8`, as an attachment, with no cache) and it throws if the headers were already sent and does not end the script; `$sendHeaders = false` only writes the content.

Errors

The failures are `RuntimeException`s (a `LogicException` when `write()` can not write the file), with the file and the reason in the message.

Validating

`validate()` says whether the file exists and can be read, whether its first `$sampleSize` rows can be read, and whether they all have the same number of columns. It does not say that the content is right, only that it looks like a CSV with that separator.

Last updated on 05/11/2026