

Introduction

Essential PHP utilities of Derafu for strings, arrays, dates, files, CSV, objects, IP addresses and URL paths

Anonymous · 08/10/2026

Derafu: Support

[github](#) [derafu/support](#) [last commit](#) [today](#)  [CI](#) [passing](#) [code size](#) [322.4 KiB](#) [open issues](#) [0](#)

[downloads](#) [8.55 k](#) [downloads](#) [1.42 k this month](#)

A collection of essential PHP utility classes for the common work of an application: strings, arrays, dates, files, CSV, objects, IP addresses and URL paths.

Every class is `final` and has only static methods. They keep no state: what a call does depends on its arguments (and on the clock or the file system, in the classes whose job is dates or files), so it gives the same answer anywhere and is easy to test.

The classes

Class	What it is for	Page
Str	Placeholders, case conversion, slugs, multibyte padding, UUIDs and random strings.	Strings
Caster	Turns a text into the scalar that it says ("42" is 42).	Strings
Encoding	UTF-8 and ISO-8859-1, also inside arrays and objects.	Strings
Arr	Dot notation, casting, trees, tables and subsets.	Arrays
Date	Spanish formatting, periods (YYYYMM), working days and units of time.	Dates
File	Atomic writes, MIME types, ZIP and downloads.	Files and CSV
Csv	Reading, writing, generating and sending CSV.	Files and CSV
Obj , Factory , Hydrator	Create objects from arrays and look at them.	Objects
Serialization , JsonSerializer	Is it serializable, and objects to JSON.	Objects
Debug	Information about a variable and where it was inspected.	Objects

Class	What it is for	Page
<code>Ip</code>	IPv4 and IPv6: validation, normalization, ranges and networks.	Ip
<code>Url</code>	The canonical form of the paths of URLs, and prefixes by segments.	Url

Why Derafu\Support?

It solves needs of business and data processing that the standard library leaves out:

- **Business dates:** working days, holidays, periods (YYYYMM) and ranges that skip weekends.
- **CSV across systems:** separators, enclosures and encodings.
- **Data transformations:** trees, tables and lists, without losing the data.
- **Safe files:** atomic writes, errors that say what failed, and the MIME type of a file.
- **Addresses that are handled right:** IPv6, ranges and networks, with the same canonical form for the same address.

Install

```
composer require derafu/support
```

Only `derafu/translation` is required (see [Errors](#)). The libraries that some classes use are **not** required, because a project that does not use those classes does not need them. Install what you use:

Package	Needed by
<code>nesbot/carbon</code>	<code>Date</code> (all of it).
<code>league/csv</code> (9.27 or later)	<code>Csv</code> (all of it).
<code>symfony/mime</code>	<code>File::mimetype()</code> , and <code>File::send()</code> for the type of the file it sends.
<code>symfony/filesystem</code>	<code>File::rmdir()</code> , which does not follow symbolic links.
<code>maennchen/zipstream-php</code>	<code>File::zip()</code> and <code>File::compress()</code> .
<code>ext-zip</code>	<code>File::unzip()</code> .

The string classes (`Str`, `Encoding`) use the functions of `mbstring`.

Errors

The errors that the classes throw are **translatable**: they are the standard exceptions of PHP (`InvalidArgumentException`, `RuntimeException`, `LogicException`) in the translatable version of [derafu/translation](#), so catching the standard class works, and the message is also available as a

message id with its parameters.

The Spanish translations ship with the package. To have them in your translator, import its services (or add `SupportTranslationResourceProvider` to the resource providers of the translator):

```
imports:
  - { resource: '../vendor/derafu/support/resources/config/support-services.yaml' }
```

`derafu/foundation` already does it.

Last updated on 08/10/2026