# Directories

Create, list and remove directories with Directory, without following symbolic links

Anonymous · 05/11/2026

## Directories

```
use Derafu\File\Directory;
```

| Method | What it does |
|---|---|
| `Directory::ensure($path, $mode = 0777)` | Makes sure that a directory exists. |
| `Directory::files($path, $extensions = [], $recursive = true)` | The files that a directory has. |
| `Directory::remove($path)` | Removes a directory with everything inside. |

A symbolic link is never followed: a link to a directory is not entered when listing, and it is removed as a link when removing.

### Ensure

```
Directory::ensure('/var/app/cache/twig');
Directory::ensure('/var/app/private', 0700);
```

The directories that are missing are created, and one that exists is left as it is. The mode is the one of the directories that are created, before the `umask`. Something that is not a directory where it must be created, or a creation that fails, is a `FileException`. Another process that creates the same directory at the same time is not an error.

### Files

```
Directory::files('/var/app/templates');
// ['/var/app/templates/a.twig', '/var/app/templates/emails/b.twig']

Directory::files('/var/app/templates', ['twig', '.TXT'], recursive: false);
```

Only files are listed (not directories), with their hidden ones, in the order of their paths. Each path starts with the directory as it was given, without the separators at the end. Symbolic links to files are

listed; the files inside a symbolic link to a directory are not.

The extensions are written with or without the point and in any case; with none, every file is listed. A directory without files is the empty list.

| Case | Exception |
| --- | --- |
| The directory does not exist. | `FileNotFoundException` |
| The path is not a directory, or it can not be read. | `FileAccessException` |
| A directory inside can not be read. | `FileException` |
| The path has a null character. | `PathException` |

## Remove

```
Directory::remove('/tmp/work');
```

It removes the directory and everything inside. A symbolic link inside is removed, and what it points to stays; a symbolic link to a directory is removed as a link. A directory that does not exist is already removed: nothing happens.

A file is a `FileAccessException`. Something that can not be removed (for instance a directory without permission) is a `FileException` whose message names it and gives the reason.