

# setup\_logging

---

## 1. Introduction

This setup function provides a clean logging setup for the `log` crate using `fern`.

- Messages with linebreaks are properly indented.
- Timestamps are only printed if they changed from timestamp of previous line.

Console only:

- “\r” at the beginning of a message overwrites previous line.
- Logging levels are colour-coded.

## 2. Table of Contents

1. Introduction .....	1
2. Table of Contents .....	1
3. Installation .....	2
4. Usage .....	2
4.1. Example .....	2
4.2. Example .....	2

## 3. Installation

1. `cargo add setup_logging --git https://github.com/9-FS/setup_logging`
2. In your `Cargo.toml` entry, replace "version" with "tag" to select the version you want to use.

### ⓘ Info

Cargo does not support automatic versioning for GitHub dependencies. Manual updates are required in the `Cargo.toml` file using "tag".

## 4. Usage

Call `setup_logging` with the desired minimum log level and log filepath format according to the `chrono` crate. After that, `log` can be used as usual.

### 4.1. Example

```
<> rust
setup_logging::setup_logging(log::Level::Info, "./log/%Y-%m-%d.log");

log::debug!("debug message"); // not printed
log::info!("info message"); // printed
log::warn!("warn message"); // printed without timestamp
log::error!("error message"); // printed without timestamp
```

### 4.2. Example

```
<> rust
setup_logging::setup_logging(log::Level::Debug, "./log/%Y-%m-%dT%H_%M.log");

log::debug!("debug message"); // printed
log::info!("info message"); // printed without timestamp
log::warn!("warn message"); // printed without timestamp
log::error!("error message"); // printed without timestamp
```