

Logging Handler

Rich supplies a logging handler which will format and colorize text written by Python's logging module.

Here's an example of how to set up a rich logger:

Rich logs won't render Console Markup in logging by default as most libraries won't be aware of the need to escape literal square brackets, but you can enable it by setting `markup=True` on the handler. Alternatively you can enable it per log message by supplying the extra argument as follows:

Similarly, the highlighter may be overridden per log message:

Handle exceptions

The `RichHandler` class may be configured to use Rich's `Traceback` class to format exceptions, which provides more context than a built-in exception. To get beautiful exceptions in your logs set `rich_tracebacks=True` on the handler constructor:

There are a number of other options you can use to configure logging output, see the `RichHandler` reference for details.

Suppressing Frames

If you are working with a framework (click, django etc), you may only be interested in seeing the code from your own application within the traceback. You can exclude framework code by setting the `suppress` argument on `Traceback`, `install`, and `Console.print_exception`, which should be a list of modules or str paths.

Here's how you would exclude click from Rich exceptions:

Suppressed frames will show the line and file only, without any code.