NAME

Log::Any::Adapter::Screen − Send logs to screen, with colors and some other features

VERSION

This document describes version 0.140 of Log::Any::Adapter::Screen (from Perl distribution Log-Any-Adapter-Screen), released on 2018−12−22.

SYNOPSIS

```
 use Log::Any::Adapter;
 Log::Any::Adapter->set('Screen',
     # min_level => 'debug', # default is 'warning'
     # colors    => { trace => 'bold yellow on_gray', ... }, # customize colors
     # use_color => 1, # force color even when not interactive
     # stderr    => 0, # print to STDOUT instead of the default STDERR
     # formatter => sub { "LOG: $_[1]" }, # default none
 );
```

DESCRIPTION

This Log::Any adapter prints log messages to screen (STDERR/STDOUT). The messages are colored according to level (unless coloring is turned off). It has a few other features: allow passing formatter, allow setting level from some environment variables, add prefix/timestamps.

Parameters:

- min_level => STRING

  Set logging level. Default is warning. If LOG_LEVEL environment variable is set, it will be used instead. If TRACE environment variable is set to true, level will be set to 'trace'. If DEBUG environment variable is set to true, level will be set to 'debug'. If VERBOSE environment variable is set to true, level will be set to 'info'.If QUIET environment variable is set to true, level will be set to 'error'.

- use_color => BOOL

  Whether to use color or not. Default is true only when running interactively (−t STDOUT returns true).

- colors => HASH

  Customize colors. Hash keys are the logging methods, hash values are colors supported by Term::ANSIColor.

  The default colors are:

method/level	color
trace	yellow
debug	(none, terminal default)
info, notice	green
warning	bold blue
error	magenta
critical, alert, emergency	red

- stderr => BOOL

  Whether to print to STDERR, default is true. If set to 0, will print to STDOUT instead.

- formatter => CODEREF

  Allow formatting message. If defined, message will be passed before being colorized. Coderef will be passed:

  ```
  ($self, $message)
  ```

  and is expected to return the formatted message.

The default formatter can optionally prefix the message with extra stuffs, depending on the content of LOG_PREFIX environment variable, such as: elapsed time (e.g. `[0.023ms]`) if LOG_PREFIX is `elapsed`.

NOTE: Log::Any 1.00+ now has a proxy object which allows formatting/customization of message before it is sent to adapter(s), so formatting does not have to be done on a per-adapter basis. As an alternative to this attribute, you can also consider using the proxy object or the (upcoming?) global proxy object.

- default_level => STR (default: warning)

  If no level-setting environment variables are defined, will default to this level.

ENVIRONMENT

NO_COLOR

If defined, will disable color. Consulted before "COLOR".

COLOR

Can be set to 0 to explicitly disable colors. The default is to check for `<-t STDOUT>`.

LOG_LEVEL => str
QUIET => bool
VERBOSE => bool
DEBUG => bool
TRACE => bool

These environment variables can set the default for `min_level`. See documentation about `min_level` for more details.

LOG_PREFIX => str

The default formatter groks these variables. See documentation about `formatter` about more details.

HOMEPAGE

Please visit the project's homepage at <https://metacpan.org/release/Log−Any−Adapter−Screen>.

SOURCE

Source repository is at <https://github.com/perlancar/perl−Log−Any−Adapter−ScreenColoredLevel>.

BUGS

Please report any bugs or feature requests on the bugtracker website <https://rt.cpan.org/Public/Dist/Display.html?Name=Log−Any−Adapter−Screen>

When submitting a bug or request, please include a test-file or a patch to an existing test-file that illustrates the bug or desired feature.

SEE ALSO

Originally inspired by Log::Log4perl::Appender::ScreenColoredLevel. The old name for this adapter is Log::Any::Adapter::ScreenColoredLevel but at some point I figure using a shorter name is better for my fingers.

Log::Any

Log::Log4perl::Appender::ScreenColoredLevel

Term::ANSIColor

AUTHOR

perlancar <perlancar@cpan.org>

COPYRIGHT AND LICENSE