

NAME

Log::Any::Test – Test what you’re logging with Log::Any

VERSION

version 1.708

SYNOPSIS

```
use Test::More;
use Log::Any::Test;    # should appear before 'use Log::Any!'
use Log::Any qw($log);

# ...
# call something that logs using Log::Any
# ...

# now test to make sure you logged the right things

$log->contains_ok(qr/good log message/, "good message was logged");
$log->does_not_contain_ok(qr/unexpected log message/, "unexpected message was no
$log->empty_ok("no more logs");

# or

my $msgs = $log->msgs;
cmp_deeply($msgs, [{message => 'msg1', level => 'debug'}, ...]);
```

DESCRIPTION

Log::Any::Test is a simple module that allows you to test what has been logged with Log::Any. Most of its API and implementation have been taken from Log::Any::Dispatch.

Using Log::Any::Test signals Log::Any to send all subsequent log messages to a single global in-memory buffer and to make the log proxy provide a number of testing functions. To use it, load Log::Any::Test before anything that loads Log::Any. To actually use the test methods, you need to load Log::Any and get a log object from it, as shown in the “SYNOPSIS”.

METHODS

The `test_name` is optional in the `*_ok` methods; a reasonable default will be provided.

`msgs()`

Returns the current contents of the global log buffer as an array reference, where each element is a hash containing a *category*, *level*, and *message* key. e.g.

```
{
  category => 'Foo',
  level => 'error',
  message => 'this is an error'
},
{
  category => 'Bar::Baz',
  level => 'debug',
  message => 'this is a debug'
}
```

`contains_ok($regex[, $test_name])`

Tests that a message in the log buffer matches *\$regex*. On success, the message is *removed* from the log buffer (but any other matches are left untouched).

`does_not_contain_ok($regex[, $test_name])`

Tests that no message in the log buffer matches *\$regex*.

`category_contains_ok ($category, $regex[, $test_name])`

Tests that a message in the log buffer from a specific category matches `$regex`. On success, the message is *removed* from the log buffer (but any other matches are left untouched).

`category_does_not_contain_ok ($category, $regex[, $test_name])`

Tests that no message from a specific category in the log buffer matches `$regex`.

`empty_ok ([$test_name])`

Tests that there is no log buffer left. On failure, the log buffer is cleared to limit further cascading failures.

`contains_only_ok ($regex[, $test_name])`

Tests that there is a single message in the log buffer and it matches `$regex`. On success, the message is removed.

`clear ()`

Clears the log buffer.

SEE ALSO

Log::Any, Test::Log::Dispatch

AUTHORS

- Jonathan Swartz <swartz@pobox.com>
- David Golden <dagolden@cpan.org>
- Doug Bell <preaction@cpan.org>
- Daniel Pittman <daniel@rimspace.net>
- Stephen Thirlwall <sdt@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2017 by Jonathan Swartz, David Golden, and Doug Bell.

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.