

Package: chronicler (via r-universe)

September 7, 2024

Title Add Logging To Functions

Version 0.2.2.9999

Description Decorate functions to make them return enhanced output.

The enhanced output consists in an object of type 'chronicle' containing the result of the function applied to its arguments, as well as a log detailing when the function was run, what were its inputs, what were the errors (if the function failed to run) and other useful information. Tools to handle decorated functions are included, such as a forward pipe operator that makes chaining decorated functions possible.

License GPL (>= 3)

Imports clipr, diffobj, dplyr, ggplot2, maybe, rlang, stringr, tibble, utils

Suggests knitr, lubridate, purrr, rmarkdown, testthat, tidyr

VignetteBuilder knitr

Config/fusen/version 0.5.2

Encoding UTF-8

LazyData TRUE

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Repository <https://b-rodriques.r-universe.dev>

RemoteUrl <https://github.com/b-rodriques/chronicler>

RemoteRef HEAD

RemoteSha c36eb78f532aa786e6502d535556b23ab2f6d0ab

Contents

as_chronicle	2
avia	3
bind_record	3
check_diff	4

check_g	4
flatten_record	5
fmap_record	6
is_chronicle	6
pick	7
print.chronicle	7
purely	8
read_log	9
record	9
record_ggplot	10
record_many	11
%>=%	12

Index	13
--------------	-----------

as_chronicle	<i>Coerce an object to a chronicle object.</i>
--------------	--

Description

Coerce an object to a chronicle object.

Usage

```
as_chronicle(.x, .log_df = data.frame())
```

Arguments

.x	Any object.
.log_df	Used internally, the user does need to interact with it. Defaults to an empty data frame.

Value

Returns a chronicle object with the object as the \$value.

Examples

```
as_chronicle(3)
```

avia	<i>avia Air passenger transport between the main airports of Luxembourg and their main partner airports</i>
------	---

Description

A non-tidy dataset from EUROSTAT which can be found [here](#).

Usage

```
avia
```

Format

A data frame with 1,434 rows and 332 columns.

bind_record	<i>Evaluate a decorated function; used to chain multiple decorated functions.</i>
-------------	---

Description

Evaluate a decorated function; used to chain multiple decorated functions.

Usage

```
bind_record(.c, .f, ...)
```

Arguments

.c	A chronicle object.
.f	A chronicle function to apply to the returning value of .c.
...	Further parameters to pass to .f.

Value

A chronicle object.

Examples

```
r_sqrt <- record(sqrt)
r_exp <- record(exp)
3 |> r_sqrt() |> bind_record(r_exp)
```

check_diff	<i>Check the output of the diff column</i>
------------	--

Description

Check the output of the diff column

Usage

```
check_diff(.c, columns = c("ops_number", "function"))
```

Arguments

.c A chronicle object.
columns Columns to select for the output. Defaults to c("ops_number", "function").

Details

diff is an option argument to the record() function. When diff = "full", a diff of the input and output of the decorated function gets saved, and if diff = "summary" only a summary of the diff is saved.

Value

A data.frame with the selected columns and column "diff_obj".

Examples

```
r_subset <- record(subset, diff = "full")  
result <- r_subset(mtcars, select = am)  
check_diff(result) # <- this is the data frame listing the operations and the accompanying diffs  
check_diff(result)$diff_obj # <- actually look at the diffs
```

check_g	<i>Check the output of the .g function</i>
---------	--

Description

Check the output of the .g function

Usage

```
check_g(.c, columns = c("ops_number", "function"))
```

Arguments

`.c` A chronicle object.

`columns` Columns to select for the output. Defaults to `c("ops_number", "function")`.

Details

`.g` is an option argument to the `record()` function. Providing this optional function allows you, at each step of a pipeline, to monitor interesting characteristics of the value object. See the package's README file for an example with data frames.

Value

A data.frame with the selected columns and column "g".

Examples

```
r_subset <- record(subset, .g = dim)
result <- r_subset(mtcars, select = am)
check_g(result)
```

flatten_record	<i>Flatten nested chronicle objects</i>
----------------	---

Description

Flatten nested chronicle objects

Usage

```
flatten_record(.c)
```

Arguments

`.c` A nested chronicle object, where the `$value` element is itself a chronicle object

Value

Returns `.c` where value is the actual value, and logs are concatenated.

Examples

```
r_sqrt <- record(sqrt)
r_log <- record(log)
a <- as_chronicle(r_log(10))
a
flatten_record(a)
```

fmap_record	<i>Evaluate a non-chronicle function on a chronicle object.</i>
-------------	---

Description

Evaluate a non-chronicle function on a chronicle object.

Usage

```
fmap_record(.c, .f, ...)
```

Arguments

.c	A chronicle object.
.f	A non-chronicle function.
...	Further parameters to pass to .f.

Value

Returns the result of `.f(.c$value)` as a new chronicle object.

Examples

```
as_chronicle(3) |> fmap_record(sqrt)
```

is_chronicle	<i>Checks whether an object is of class "chronicle"</i>
--------------	---

Description

Checks whether an object is of class "chronicle"

Usage

```
is_chronicle(.x)
```

Arguments

.x	An object to test.
----	--------------------

Value

TRUE if .x is of class "chronicle", FALSE if not.

pick	<i>Retrieve an element from a chronicle object.</i>
------	---

Description

Retrieve an element from a chronicle object.

Usage

```
pick(.c, .e)
```

Arguments

.c	A chronicle object.
.e	Element of interest to retrieve, one of "value" or "log_df".

Value

The value or log_df element of the chronicle object .c.

Examples

```
r_sqrt <- record(sqrt)
r_exp <- record(exp)
3 |> r_sqrt() %>=% r_exp() |> pick("value")
```

print.chronicle	<i>Print method for chronicle objects.</i>
-----------------	--

Description

Print method for chronicle objects.

Usage

```
## S3 method for class 'chronicle'
print(x, ...)
```

Arguments

x	A chronicle object.
...	Unused.

Details

chronicle object are, at their core, lists with the following elements:

- "\$value": a an object of type maybe containing the result of the computation (see the "Maybe monad" vignette for more details on maybes).
- "\$log_df": a data.frame object containing the printed object's log information.

print.chronicle() prints the object on screen and shows:

- the value using its print() method (for example, if the value is a data.frame, print.data.frame() will be used)
- a message indicating to the user how to recuperate the value inside the chronicle object and how to read the object's log

Value

No return value, called for side effects (printing the object on screen).

purely

Capture all errors, warnings and messages.

Description

Capture all errors, warnings and messages.

Usage

```
purely(.f, strict = 2)
```

Arguments

.f	A function to decorate.
strict	Controls if the decorated function should catch only errors (1), errors and warnings (2, the default) or errors, warnings and messages (3).

Value

A function which returns a list. The first element of the list, \$value, is the result of the original function .f applied to its inputs. The second element, \$log is NULL in case everything goes well. In case of error/warning/message, \$value is NA and \$log holds the message. purely() is used by record() to allow the latter to handle errors.

Examples

```
purely(log)(10)
purely(log)(-10)
purely(log, strict = 1)(-10) # This produces a warning, so with strict = 1 nothing gets captured.
```

read_log	<i>Reads the log of a chronicle.</i>
----------	--------------------------------------

Description

Reads the log of a chronicle.

Usage

```
read_log(.c)
```

Arguments

.c A chronicle object.

Value

The log of the object.

Examples

```
## Not run:  
read_log(chronicle_object)  
  
## End(Not run)
```

record	<i>Decorates a function to output objects of type chronicle.</i>
--------	--

Description

Decorates a function to output objects of type chronicle.

Usage

```
record(.f, .g = (function(x) NA), strict = 2, diff = "none")
```

Arguments

.f A function to decorate.
.g Optional. A function to apply to the intermediary results for monitoring purposes. Defaults to returning NA.
strict Controls if the decorated function should catch only errors (1), errors and warnings (2, the default) or errors, warnings and messages (3).
diff Whether to show the diff between the input and the output ("full"), just a summary of the diff ("summary"), or none ("none", the default)

Details

To chain multiple decorated function, use `bind_record()` or `%>=%`. If the `diff` parameter is set to "full", `diffobj::diffObj()` (or `diffobj::summary(diffobj::diffObj())`, if `diff` is set to "summary") gets used to provide the diff between the input and the output. This diff can be found in the `log_df` element of the result, and can be viewed using `check_diff()`.

Value

A function which returns objects of type `chronicle`. `chronicle` objects carry several elements: a `value` which is the result of the function evaluated on its inputs and a second object called `log_df`. `log_df` contains logging information, which can be read using `read_log()`. `log_df` is a data frame with columns: `outcome`, `function`, `arguments`, `message`, `start_time`, `end_time`, `run_time`, `g` and `diff_obj`.

Examples

```
record(sqrt)(10)
```

```
record_ggplot
```

```
Record ggplot
```

Description

`record_ggplot` takes a `ggplot_expression` and an optional `strict` argument as input and does the following steps:

1. Records the `ggplot_fun` function with the given `strict` argument using the `record` function from `chronicler`
2. Passes the `ggplot_expression` to the recorded `ggplot_fun` function
3. Returns the result of the recorded `ggplot_fun` function

Usage

```
record_ggplot(ggplot_expression, strict = 2)
```

Arguments

`ggplot_expression`

A `ggplot` expression.

`strict`

An optional integer argument controlling the behavior of the `record()` function from `chronicler`. Default is 2.

Value

A `chronicler` object.

Examples

```
## Not run:
library(ggplot2)
# Unsuccessful example
x <- record_ggplot(ggplot(data = mtcars) + geom_point(aes(y = hp, x = mpg)))
print(x)

# Successful example
z <- record_ggplot(ggplot(data = mtcars) + geom_point(aes(y = hp, x = mpg)))
print(z)

## End(Not run)
```

record_many

Decorate a list of functions

Description

Decorate a list of functions

Usage

```
record_many(list_funcs, .g = (function(x) NA), strict = 2, diff = "none")
```

Arguments

<code>list_funcs</code>	A list of function names, as strings.
<code>.g</code>	Optional. Defaults to a function which returns NA.
<code>strict</code>	Controls if the decorated function should catch only errors (1), errors and warnings (2, the default) or errors, warnings and messages (3).
<code>diff</code>	Whether to show the diff between the input and the output ("full"), just a summary of the diff ("summary"), or none ("none", the default)

Details

Functions must be entered as strings of the form "function" or "package::function". The code gets generated and copied into the clipboard. The code can then be pasted into the text editor. On GNU/Linux systems, you might get the following error message on first use: "Error in : Clipboard on X11 requires that the DISPLAY envvar be configured". This is an error message from `clipr::write_clip()`, used by `record_many()` to put the generated code into the system's clipboard. To solve this issue, run `echo $DISPLAY` in the system's shell. This command should return a string like `":0"`. Take note of this string. In your `.Rprofile`, put the following command: `Sys.setenv(DISPLAY = ":0")` and restart the R session. `record_many()` should now work.

Value

Puts a string into the systems clipboard.

Examples

```
## Not run:  
list_funcs <- list("exp", "dplyr::select", "exp")  
record_many(list_funcs)  
  
## End(Not run)
```

%>=%

Pipe a chronicle object to a decorated function.

Description

Pipe a chronicle object to a decorated function.

Usage

```
.c %>=% .f
```

Arguments

.c	A value returned by record.
.f	A chronicle function to apply to the returning value of .c.

Value

A chronicle object.

Examples

```
r_sqrt <- record(sqrt)  
r_exp <- record(exp)  
3 |> r_sqrt() %>=% r_exp()
```

Index

* datasets

avia, 3

%>=%, 12

as_chronicle, 2

avia, 3

bind_record, 3

check_diff, 4

check_g, 4

flatten_record, 5

fmap_record, 6

is_chronicle, 6

pick, 7

print.chronicle, 7

purely, 8

read_log, 9

record, 9

record_ggplot, 10

record_many, 11