

Package: timeless (via r-universe)

September 17, 2024

Title Fast General Purpose Date/Time Converter

Version 0.2.4.9000

Description Fast general purpose date/time converter using 'Rust'. The package implements date time, date and epoch time parser for heterogeneous vectors of dates.

URL <https://github.com/schochastics/timeless>,
<https://schochastics.github.io/timeless/>

BugReports <https://github.com/schochastics/timeless/issues>

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Config/rextendr/version 0.3.1.9000

Depends R (>= 4.1)

LazyData true

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

SystemRequirements Cargo (Rust's package manager), rustc (>= 1.67.1)

Repository <https://schochastics.r-universe.dev>

RemoteUrl <https://github.com/schochastics/timeless>

RemoteRef HEAD

RemoteSha 79479b7611cb953601375f74c3e2aaab6c03dfb7

Contents

bench_date	2
chronos	2
parse_date	3
parse_datetime	3
parse_epoch	4

Index**5**

bench_date	<i>A benchmark dataset with different date formats</i>
------------	--

Description

A benchmark dataset with different date formats

Usage

```
bench_date
```

Format

An object of class character of length 93.

chronos	<i>Fast general purpose parser for date(time) from input data</i>
---------	---

Description

Fast general purpose parser for date(time) from input data

Usage

```
chronos(x, formats = NULL, tz = "", to_tz = "", out_format = "datetime")
```

Arguments

x	A vector with date(time) expressions to be parsed and converted.
formats	character vector of formats to try out (see base::strptime). If NULL, uses a set of predefined formats mostly taken from the anytime package.
tz	assumed input timezone. If "", uses local timezone. See details
to_tz	convert datetime to timezone given in to_tz. If "", tz is used. See details
out_format	character. Defining the format of the returned result. Can be "datetime", "date", or "character".

Details

The internal parsing is done "timezoneless". The timezone given in tz is just added to the datetime without any conversion. If to_tz is given, a conversion is made from tz to to_tz.

Value

A character vector which can be transformed to POSIXct or date

See Also

[parse_datetime](#), [parse_date](#), and [parse_epoch](#) if you need more control over formatting

Examples

```
chronos(bench_date)
```

parse_date	<i>Parse date from strings using different formats</i>
------------	--

Description

Parse date from strings using different formats

Usage

```
parse_date(x, formats = NULL, out_date = "%Y-%m-%d")
```

Arguments

x	A vector with date(time) expressions to be parsed and converted.
formats	character vector of formats to try out (see base::strptime). If NULL, uses a set of predefined formats mostly taken from the anytime package.
out_date	character defining the date format of the parsed strings

Value

character vector of parsed dates.

parse_datetime	<i>Parse datetime from strings using different formats</i>
----------------	--

Description

Parse datetime from strings using different formats

Usage

```
parse_datetime(x, formats = NULL, out_datetime = "%Y-%m-%d %H:%M:%S")
```

Arguments

x	A vector with date(time) expressions to be parsed and converted.
formats	character vector of formats to try out (see base::strptime). If NULL, uses a set of predefined formats mostly taken from the anytime package.
out_datetime	character defining the datetime format of the parsed strings

Value

character vector of parsed datetimes

parse_epoch	<i>Parse datetime from epoch</i>
-------------	----------------------------------

Description

Parse datetime from epoch

Usage

```
parse_epoch(x, tz = "", out_datetime = "%Y-%m-%d %H:%M:%S")
```

Arguments

x	A vector with date(time) expressions to be parsed and converted.
tz	timezone of output datetime. If "", uses local timezone
out_datetime	character defining the datetime format of the parsed strings

Value

character vector of parsed dates.

Index

* **datasets**

bench_date, [2](#)

base::strptime, [2](#), [3](#)

bench_date, [2](#)

chronos, [2](#)

parse_date, [3](#), [3](#)

parse_datetime, [3](#), [3](#)

parse_epoch, [3](#), [4](#)