Package ‘tibblify’

September 23, 2020

Title Rectangle Nested Lists
Version 0.1.0
Description A tool to rectangle a nested list, that is to convert it into a tibble. This is done automatically or according to a given specification. A common use case is for nested lists coming from parsing JSON files or the JSON response of REST APIs. It is supported by the 'vctrs' package and therefore offers a wide support of vector types.
License GPL-3
Suggests testthat, knitr, rmarkdown, covr, spelling
Encoding UTF-8
LazyData true
RoxygenNote 7.1.1
Imports crayon, rlang, vctrs, purrr, tibble
VignetteBuilder knitr
Depends R (>= 3.4.0)
URL https://github.com/mgirlich/tibblify
BugReports https://github.com/mgirlich/tibblify/issues
Language en-US
NeedsCompilation no
Author Maximilian Girlich [aut, cre]
Maintainer Maximilian Girlich <maximilian.girlich@outlook.com>
Repository CRAN
Date/Publication 2020-09-23 08:40:07 UTC

R topics documented:

get_spec ................................................................. 2
got_chars ............................................................... 2
lcols ................................................................. 3
get_spec

Description
Examine the column specification

Usage
get_spec(x)

Arguments
x The data frame object to extract from

got_chars

Game of Thrones POV characters

Description
The data is from the repurrsive package.

Usage
got_chars

Format
A unnamed list with 30 components, each representing a POV character. Each character’s component is a named list of length 18, containing information such as name, aliases, and house allegiances.

Details
Info on the point-of-view (POV) characters from the first five books in the Song of Ice and Fire series by George R. R. Martin. Retrieved from An API Of Ice And Fire.

Source
https://anapioficeandfire.com
**Examples**

```r
got_chars
str(lapply(got_chars, `\[`, c("name", "culture")))
```

---

**lcols** \hspace{1cm} *Create list parser specification*

**Description**

Create list parser specification

**Usage**

```r
lcols(..., .default = lcol_skip(zap()))
```

**Arguments**

- `...`: Column specification objects created by `lcol_*(.)`
- `.default`: The default parser to use.

---

**lcol_lgl** \hspace{1cm} *Create column specification*

**Description**

`lcols()` includes all fields in the input data, skipping the column types as the default.

**Usage**

```r
lcol_lgl(path, .default = zap(), .parser = NULL)
lcol_int(path, .default = zap(), .parser = NULL)
lcol_dbl(path, .default = zap(), .parser = NULL)
lcol_chr(path, .default = zap(), .parser = NULL)
lcol_dat(path, .default = zap(), .parser = NULL)
lcol_dtt(path, .default = zap(), .parser = NULL)
lcol_vec(path, ptype, .default = zap(), .parser = NULL)
lcol_guess(path, .default = NULL)
```
lcol_skip(path)
lcol_lst(path, .default = zap(), .parser = NULL)
lcol_lst_of(path, .ptype, .default = zap(), .parser = NULL)
lcol_df_lst(path, ..., .default = zap())
lcol_df(path, ..., .default = zap())

Arguments

path  A character vector or list that is converted to an extractor function (similar to the
      .f argument in purrr::map()).
.default  Value to use if target is empty or absent. If zap() (the default) an error is thrown
          if target is empty or absent.
.parser  A transformation applied to each element of the list before coercing to .ptype.
          This is usually needed for lcol_dat() and lcol_dtt().
.ptype   The prototype of the vector.
.ptype   The .ptype for vctrs::list_of().
...  Column specification passed on to lcols().

politicians  

Politicians

Description

A dataset containing some basic information about some politicians.

Usage

politicians

Format

A list of lists.
**tibblify**

*Rectange a nested list into a tidy tibble*

---

**Description**

Rectange a nested list into a tidy tibble

**Usage**

```r
tibblify(recordlist, col_specs = lcols(.default = lcol_guess(zap())))
```

**Arguments**

- `recordlist`: A nested list.
- `col_specs`: A specification generated with `lcols()` how to turn the list into a tibble.

**Value**

The tibble generated according to the specification.

**Examples**

```r
recordlist <- list(
  list(id = 1, name = "Tyrion Lannister"),
  list(id = 2, name = "Victarion Greyjoy")
)
tibblify(recordlist)
```
Index

* Game of Thrones data and functions
  got_chars, 2
* datasets
  got_chars, 2
  politicians, 4
get_spec, 2
got_chars, 2
lcol_chr (lcol_lgl), 3
lcol_dat (lcol_lgl), 3
lcol_db1 (lcol_lgl), 3
lcol_df (lcol_lgl), 3
lcol_df_lst (lcol_lgl), 3
lcol_dtt (lcol_lgl), 3
lcol_guess (lcol_lgl), 3
lcol_int (lcol_lgl), 3
lcol_lgl, 3
lcol_lst (lcol_lgl), 3
lcol_lst_of (lcol_lgl), 3
lcol_skip (lcol_lgl), 3
lcol_vec (lcol_lgl), 3
lcols, 3
politicians, 4
tibblify, 5