

Package ‘xlcutter’

July 21, 2025

Title Parse Batches of 'xlsx' Files Based on a Template

Version 0.1.1

Description Parse entire folders of non-rectangular 'xlsx' files into a single rectangular and tidy 'data.frame' based on a custom template file defining the column names of the output.

License MIT + file LICENSE

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

RoxygenNote 7.2.3

Depends R (>= 3.3.0)

Imports tidyxl

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

URL <https://github.com/Bisaloo/xlcutter>,
<https://hugogruson.fr/xlcutter/>

BugReports <https://github.com/Bisaloo/xlcutter/issues>

NeedsCompilation no

Author Hugo Gruson [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0002-4094-1476>>)

Maintainer Hugo Gruson <hugo.gruson+R@normalesup.org>

Repository CRAN

Date/Publication 2023-10-22 19:00:02 UTC

Contents

validate_xltemplate	2
xlsx_cutter	3

Index	5
--------------	----------

validate_xltemplate	<i>Validate an xlsx template file to use in xlsx_cutter()</i>
---------------------	---

Description

Validate an xlsx template file to use in [xlsx_cutter\(\)](#)

Usage

```
validate_xltemplate(  
  template_file,  
  template_sheet = 1,  
  marker_open = "{{",  
  marker_close = "}}",  
  minimal = FALSE,  
  error = FALSE  
)
```

Arguments

template_file	path to the template file to use as a model to parse the xlsx files in data_folder
template_sheet	sheet id of the template file to use as a model to parse the xlsx files in data_folder
marker_open, marker_close	character marker to mark the variables to extract in the template_file
minimal	Logical (default to FALSE) saying whether the template should contain only variables delimited by markers and nothing else, or if extra text can be included (and ignored)
error	Logical (defaults to TRUE) saying whether failed validations should result in an error (TRUE) or a warning (FALSE)

Value

TRUE if the template is valid, FALSE otherwise

Examples

```
# Valid template
validate_xltemplate(  
  system.file("example", "timesheet_template.xlsx", package = "xlcutter")  
)  
  
# Invalid templates
validate_xltemplate(  
  system.file("example", "template_duped_vars.xlsx", package = "xlcutter")  
)  
  
validate_xltemplate(  
  system.file("example", "template_duped_vars.xlsx", package = "xlcutter")  
)
```

```

    system.file("example", "template_fluff.xlsx", package = "xlcutter"),
    minimal = TRUE
  )

```

xlsx_cutter

*Create a data.frame from a folder of non-rectangular excel files***Description**

Create a data.frame from a folder of non-rectangular excel files based on a defined custom template

Usage

```

xlsx_cutter(
  data_files,
  template_file,
  data_sheet = 1,
  template_sheet = 1,
  marker_open = "{{",
  marker_close = "}}"
)

```

Arguments

data_files vector of paths to the xlsx files to parse

template_file path to the template file to use as a model to parse the xlsx files in data_folder

data_sheet sheet id to extract from the xlsx files

template_sheet sheet id of the template file to use as a model to parse the xlsx files in data_folder

marker_open, marker_close

 character marker to mark the variables to extract in the template_file

Value

A rectangular data.frame with columns as defined in the template. Column types are determined automatically by type.convert()

Examples

```

data_files <- list.files(
  system.file("example", "timesheet", package = "xlcutter"),
  pattern = "\\..xlsx$",
  full.names = TRUE
)

template_file <- system.file(
  "example", "timesheet_template.xlsx",
  package = "xlcutter"
)

```

```
)  
  
xlsx_cutter(  
    data_files,  
    template_file  
)
```

Index

`validate_xltemplate`, [2](#)

`xlsx_cutter`, [3](#)

`xlsx_cutter()`, [2](#)