Package 'rtiddlywiki'

May 9, 2026

Type Package

Title R Interface for 'TiddlyWiki'

Version 0.5.2

Description 'TiddlyWiki' is a unique non-linear notebook for capturing, organising and sharing complex information. 'rtiddlywiki' is a R interface of 'TiddlyWiki' <https://tiddlywiki.com> to create new tiddler from 'R Markdown' file, and then put into a local 'TiddlyWiki' server if it is available.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

URL https://rtiddlywiki.bangyou.me/, https://github.com/byzheng/rtiddlywiki

BugReports https://github.com/byzheng/rtiddlywiki/issues

Imports settings, stats, httr2 (>= 1.2.2), pingr, websocket, rmarkdown, later, utils, jsonlite, bookdown, stringr, stringi, digest, htmlwidgets, rvest, knitr, base64enc, htmltools, grDevices

RoxygenNote 7.3.3

Suggests leaflet, dplyr, testthat (>= 3.0.0), processx, ggplot2, grid

NeedsCompilation no

Author Bangyou Zheng [aut, cre]

Maintainer Bangyou Zheng <bangyou.zheng@csiro.au>

Repository CRAN

Date/Publication 2025-12-18 08:00:02 UTC

Contents

`create_tabs` *Create HTML Tabs for Multiple Objects*

Description

This function generates a tabbed HTML interface for displaying multiple R objects such as ‘ggplot2‘ plots and data frames. Each tab can include custom arguments for rendering (e.g., image width or table formatting).

Usage

```
create_tabs(...)
```

Arguments

...	Named arguments where each name defines a tab label. Each argument can be either: • A single object (‘ggplot‘, ‘data.frame‘, etc.) • A list with an ‘object‘ element and additional named arguments passed to the relevant rendering function.

Details

Supported object types:

- ‘ggplot‘ objects — rendered as base64-encoded images using ‘save_base64()‘
- ‘data.frame‘ or ‘tibble‘ — rendered using ‘kable_html()‘

Additional arguments passed inside the list are used by the relevant rendering function.

Value

A ‘htmltools::tag‘ object representing the full tab interface. Can be printed in R Markdown documents or displayed interactively in RStudio Viewer.

Examples

```
## Not run:
library(ggplot2)

# Simple plot and table
p <- ggplot(cars) + geom_point(aes(speed, dist))
df <- head(cars)

# Basic usage
create_tabs(
  plot = p,
  table = df
)

# With custom rendering arguments
create_tabs(
  plot = list(object = p, width = 4),
  table = list(object = df, digits = 2)
)

## End(Not run)
```

---

delete_tiddler *Delete a tiddler*

Description

Delete a tiddler

Usage

```
delete_tiddler(title, bag = TW_OPTIONS("bag"))
```

Arguments

title	title of the tiddler to retrieve
bag	string defining which recipe to write to (optional, defaults to "default")

Value

no return values

Examples

```
## Not run:
delete_tiddler("GettingStarted")

## End(Not run)
```

---

get_status *Get server status*

Description

Get server status

Usage

```
get_status()
```

Value

a list of service status

---

get_tiddler *Get a tiddler*

Description

Get a tiddler

Usage

```
get_tiddler(title, recipe = TW_OPTIONS("recipe"))
```

Arguments

title	title of the tiddler to retrieve
recipe	string defining which recipe to read from (optional, defaults to "default")

Value

tiddler information in JSON format

Examples

```
## Not run:
get_tiddler("GettingStarted")

## End(Not run)
```

`get_tiddlers` *Get all tiddlers*

Description

Get all tiddlers

Usage

```
get_tiddlers(filter = NULL, exclude = NULL, recipe = TW_OPTIONS("recipe"))
```

Arguments

`filter`	filter identifying tiddlers to be returned (optional, defaults to "[all[tiddlers]!is[system]sort[title]]")
`exclude`	comma delimited list of fields to excluded from the returned tiddlers (optional, defaults to "text")
`recipe`	string defining which recipe to read from (optional, defaults to "default")

Value

all tiddlers information in JSON format

Examples

```
## Not run:
#' Get all tiddlers
get_tiddlers()

## End(Not run)
```

kable_html *Convert Data Frame to HTML Table using kable*

Description

Convert Data Frame to HTML Table using kable

Usage

```
kable_html(df, ...)
```

Arguments

df	A data frame to be converted to an HTML table.
...	Other arguments to be passed to ‘knitr::kable‘.

Value

A htmltools object containing the HTML representation of the table.

Examples

```
kable_html(cars[1:10,])
```

print.tw_html *Print tw_html object*

Description

Print tw_html object

Usage

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

Arguments

x	a ‘tw_html‘ object.
...	additional arguments passed to or from other methods.

Value

no return

put_tiddler *Put a tiddler*

Description

Put a tiddler

Usage

```
put_tiddler(
  title,
  text,
 type = c("text/vnd.tiddlywiki", "text/x-tiddlywiki", "text/x-markdown", "text/html",
    "text/plain", "application/json"),
  tags = NULL,
  fields = NULL,
  recipe = TW_OPTIONS("recipe")
)
```

Arguments

title	tiddler title
text	tiddler text
type	tiddler type
tags	tiddler tags which is merged with existing tags
fields	a named vector for tiddler fields which is merged with existing tags
recipe	string defining which recipe to write to (optional, defaults to "default")

Value

No return value

Examples

```
## Not run:
title <- "New tiddler"
text <- c("!! Section",
          "This is a new tiddler")
type <- "text/vnd.tiddlywiki"
tags <- c("Tag1", "Tag 2")
fields <- c("F1" = "V1", "F2" = "V2")
put_tiddler(title = title,
            text = text,
            type = type,
            tags = tags,
            fields = fields)

## End(Not run)
```

read_table *Read a TiddlyWiki Table into a Data Frame*

Description

This function parses a table written in TiddlyWiki format and converts it into an R data frame. It can optionally treat the first row as a header.

Usage

```
read_table(table, header = TRUE)
```

Arguments

table	A character string representing the TiddlyWiki table.
header	A logical value indicating whether the first row should be treated as column headers. Default is TRUE.

Value

A data frame containing the parsed table data.

Examples

```
table <- "|!Cell1 |!Cell2 |\n|Cell3 |Cell4 |"
df <- read_table(table, header = TRUE)
print(df)
```

remove_fields *Remove fields from tiddlers*

Description

Remove fields from tiddlers

Usage

```
remove_fields(title, fields, recipe = TW_OPTIONS("recipe"))
```

Arguments

title	tiddler title
fields	fields to remove
recipe	string defining which recipe to write to (optional, defaults to "default")

Value

no return value

`save_base64` *Save ggplot into base64*

Description

Save ggplot into base64

Usage

```
save_base64(plot, width = NULL, height = NULL, dpi = NULL, ...)
```

Arguments

`plot`	object for ggplot2 or a function for plot
`width`	image width
`height`	image height
`dpi`	image resolution
`...`	Other arguments for plot function

Value

character string for base64 image

Examples

```
## Not run:
library(ggplot2)
p <- cars |>
    ggplot() +
    geom_point(aes(speed, dist))
p |> save_base64()

## End(Not run)
```

split_field *Split tiddlywiki field into values*

Description

Split tiddlywiki field into values

Usage

```
split_field(s)
```

Arguments

s	a string

Value

an vector of values

tabs *Generate HTML Tabs with Dynamic Content*

Description

This function creates a tabbed interface where each tab has dynamically generated content.

Usage

```
tabs(names, fun, groupname = .unique_name(), checked = 1, ...)
```

Arguments

names	A character vector of tab labels.
fun	A function that generates the content for each tab. It must take an index (‘i‘) as the first argument.
groupname	A unique string to group the radio inputs (default is generated automatically).
checked	The index of the tab that should be pre-selected (default is ‘1‘).
...	Additional arguments passed to ‘fun‘.

Value

An ‘htmltools::tagList‘ containing the tabbed interface.

Examples

```
## Not run:
tab_labels <- c("Tab1", "Tab2", "Tab3")

tab_content_fun <- function(i, extra_text = "") {
    htmltools::tagList(
        htmltools::tags$p(paste("Content for tab:", tab_labels[i], extra_text)),
        htmltools::tags$img(src = paste0("plot_", i, ".png"), width = "100%")
    )
}

tabs(tab_labels, tab_content_fun, checked = 2, extra_text = "Additional details")

## End(Not run)
```

---

tiddler_document *Format for converting from R Markdown to another tiddler markdown*

Description

Format for converting from R Markdown to another tiddler markdown

Usage

```
tiddler_document(
  host = NULL,
  remote = FALSE,
  preview = FALSE,
  tags = NULL,
  fields = NULL,
  use_bookdown = FALSE,
  overwrite = FALSE,
  variant = "gfm",
  pandoc_args = "--wrap=none",
  ...
)
```

Arguments

host	the host of tiddlywiki web server
remote	whether put into remote TiddlyWiki Node.js Server
preview	whether to send ‘open_tiddler‘ command to ws server (tw-livebridge) to preview in browser
tags	tiddler tags
fields	a named vector for tiddler fields
use_bookdown	logical. Use bookdown to generate markdown file.

overwrite	whether to overwrite the existing tiddler.
variant	variant for md_document
pandoc_args	pandoc_args for md_document
...	Other argument pass to md_document

Value

R Markdown output format to pass to render()

Examples

```
## Not run:
library(rmarkdown)
render("input.Rmd")

## End(Not run)
```

---

tiddler_json *Generate tiddler in json format*

Description

Generate tiddler in json format

Usage

```
tiddler_json(
  title,
  text,
  type = c("text/vnd.tiddlywiki", "text/x-tiddlywiki", "text/x-markdown", "text/html",
    "text/plain", "application/json"),
  tags = NULL,
  fields = NULL,
  format = c("json", "list")
)
```

Arguments

title	tiddler title
text	tiddler text
type	tiddler type
tags	a vector for tiddler tags
fields	a named vector for tiddler fields.
format	export format as json or list

Value

New tiddler in json format

tiddler_json2 *Generate tiddler in json format*

Description

Generate tiddler in json format

Usage

```
tiddler_json2(tiddler)
```

Arguments

tiddler	A list for new tiddler

Value

New tiddler in json format

tw_options *Set or get options for my package*

Description

Set or get options for my package

Usage

```
tw_options(...)
```

Arguments

...	Option names to retrieve option values or [key]=[value] pairs to set options.

Value

the default and modified options.

Supported options

The following options are supported host: Host of tiddlywiki recipe: Recipes are named lists of bags, ordered from lowest priority to highest bag: Bags have access controls that determines which users can read or write to them

Examples

```
tw_options(host = "http://127.0.0.1:8080/")
```

tw_reset *Reset global options for pkg*

Description

Reset global options for pkg

Usage

```
tw_reset()
```

Value

the default options

Examples

```
tw_options()
```

tw_table *Convert data.frame into table of TiddlyWiki*

Description

Convert data.frame into table of TiddlyWiki

Usage

```
tw_table(df, collapse = "\n")
```

Arguments

df	data.frame object
collapse	an optional character string to separate the results.

Value

character string for table in TiddlyWiki

Examples

```
cars |>
    dplyr::slice(1:10) |>
    tw_table()
```

tw_widget *Create a tiddlywiki widget from htmlwidget*

Description

Create a tiddlywiki widget from htmlwidget

Usage

```
tw_widget(widget, is_cat = FALSE)
```

Arguments

widget	an object of htmlwidget
is_cat	whether to show results on screen

Value

a new tiddlywiki widget

Examples

```
library(leaflet)
## Not run:
content <- paste(sep = "<br/>",
                 "<b><a href='http://www.samurainoodle.com'>Samurai Noodle</a></b>",
                 "606 5th Ave. S",
                 "Seattle, WA 98138"
)

widget <- leaflet() %>% addTiles() %>%
    addPopups(-122.327298, 47.597131, content,
                 options = popupOptions(closeButton = FALSE)
    )
tw_widget(widget)

## End(Not run)
```

wikitext_backticks_return *backticks*

Description

backticks for wikitext

Usage

```
wikitext_backticks_return(x)
```

Arguments

x	character vector

Value

By default this function outputs (see: `cat`) the result. Call the function ending in `.return` to catch the result instead.

Examples

```
wikitext_backticks('FOO')
wikitext_backticks_return('FOO')
```

---

word_to_md *Convert a Word File to Markdown with Optional Embedded Images*

Description

This function converts a Word (‘.docx‘) file to Markdown using Pandoc. Optionally, it embeds images as Base64 for a self-contained Markdown file.

Usage

```
word_to_md(
  docx_file,
  output_file = NULL,
  embed_images = FALSE,
  overwrite = FALSE
)
```

Arguments

docx_file	Path to the input Word file.
output_file	Path to the final Markdown output file. If null, the original file name.
embed_images	Logical. If ‘TRUE‘, all images will be embedded as Base64. Default is ‘FALSE‘.
overwrite	Logical. If ‘TRUE‘, Output file is overwrote. Default is ‘FALSE‘

Value

Saves a Markdown file (optionally with Base64-embedded images).

Examples

```
# Convert Word to Markdown without embedding images
## Not run:
word_to_md("input.docx", "output.md", embed_images = FALSE)

# Convert and embed images as Base64
word_to_md("input.docx", "output_embedded.md", embed_images = TRUE)

## End(Not run)
```

Index