

# Package: rtiddlywiki (via r-universe)

October 18, 2024

**Type** Package

**Title** R Interface for 'TiddlyWiki'

**Version** 0.3.0

**Author** Bangyou Zheng

**Maintainer** Bangyou Zheng <Bangyou.Zheng@csiro.au>

**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 Rmarkdown file, and then put into a local 'TiddlyWiki' node.js server if it is available.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

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

**BugReports** <https://github.com/byzheng/rtiddlywiki/issues>

**Imports** settings, httr2, rmarkdown, utils, jsonlite, bookdown, stringr, stringi, htmlwidgets, rvest, knitr

**RoxygenNote** 7.3.1

**Suggests** leaflet, ggplot2, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Repository** <https://byzheng.r-universe.dev>

**RemoteUrl** <https://github.com/byzheng/rtiddlywiki>

**RemoteRef** HEAD

**RemoteSha** 5bcdaaf8cdbc3cae790b960cb81f9de4e899140d

## Contents

delete_tiddler . . . . .	2
get_tiddler . . . . .	3
get_tiddlers . . . . .	3
put_tiddler . . . . .	4
remove_fields . . . . .	5
split_field . . . . .	6
tiddler_document . . . . .	6
tiddler_json . . . . .	7
tiddler_json2 . . . . .	8
tw_options . . . . .	8
tw_reset . . . . .	9
tw_widget . . . . .	9
wikitext_backticks_return . . . . .	10
<b>Index</b>	<b>11</b>

---

delete_tiddler	<i>Delete a tiddler</i>
----------------	-------------------------

---

### 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_tiddler	<i>Get a tiddler</i>
-------------	----------------------

---

**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	<i>Get all tiddlers</i>
--------------	-------------------------

---

**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)
```

---

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)
```

---

remove_fields	<i>Remove fields from tiddlers</i>
---------------	------------------------------------

---

## 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

---

split_field	<i>Split tiddlywiki field into values</i>
-------------	---

---

**Description**

Split tiddlywiki field into values

**Usage**

```
split_field(s)
```

**Arguments**

s                    a string

**Value**

an vector of values

---

tiddler_document	<i>Format for converting from R Markdown to another tiddler markdown</i>
------------------	--

---

**Description**

Format for converting from R Markdown to another tiddler markdown

**Usage**

```
tiddler_document(
  host = NULL,
  tags = NULL,
  fields = NULL,
  use_bookdown = FALSE,
  overwrite = FALSE,
  ...
)
```

**Arguments**

host	the host of tiddlywiki web server
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.
...	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	<i>Generate tiddler in json format</i>
--------------	--

---

**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	<i>Generate tiddler in json format</i>
---------------	--

---

**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	<i>Set or get options for my package</i>
------------	--

---

**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	<i>Reset global options for pkg</i>
----------	-------------------------------------

---

**Description**

Reset global options for pkg

**Usage**

```
tw_reset()
```

**Value**

the default options

**Examples**

```
tw_options()
```

---

tw_widget	<i>Create a tiddlywiki widget from htmlwidget</i>
-----------	---

---

**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)
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)
```

---

```
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('F00')
wikitext_backticks_return('F00')
```

# Index

`delete_tiddler`, 2

`get_tiddler`, 3  
`get_tiddlers`, 3

`put_tiddler`, 4

`remove_fields`, 5

`split_field`, 6

`tiddler_document`, 6  
`tiddler_json`, 7  
`tiddler_json2`, 8  
`tw_options`, 8  
`tw_reset`, 9  
`tw_widget`, 9

`wikitext_backticks`  
    (`wikitext_backticks_return`), 10  
`wikitext_backticks_return`, 10