Package: mailmerge (via r-universe)

October 18, 2026

Title Mail Merge Using R Markdown Documents and 'gmailr'

Version 0.2.5

Description Perform a mail merge (mass email) using the message defined in markdown, the recipients in a 'csv' file, and gmail as the mailing engine. With this package you can parse markdown documents as the body of email, and the 'yaml' header to specify the subject line of the email. Any '{}' braces in the email will be encoded with 'glue::glue()'. You can preview the email in the RStudio viewer pane, and send (draft) email using 'gmailr'.

License MIT + file LICENSE

URL https://andrie.github.io/mailmerge/, https://github.com/andrie/mailmerge

BugReports https://github.com/andrie/mailmerge/issues

Depends R (>= 3.3)

Imports commonmark, gmailr, rstudioapi, googledrive, rmarkdown, glue, magrittr, purrr, utils, fs, lifecycle, shiny, miniUI

Suggests spelling, withr, covr, knitr, testthat (>= 2.1.0), here, mockery, readr

VignetteBuilder knitr

RdMacros lifecycle

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Language en-US

Config/testthat/edition 3

Config/pak/sysreqs cmake make libuv1-dev libssl-dev zlib1g-dev

Repository https://andrie.r-universe.dev

Date/Publication 2024-03-31 17:59:59 UTC

RemoteUrl https://github.com/andrie/mailmerge

RemoteRef HEAD

RemoteSha 0f1004b8ca813ebc85e0748c1075b285b057406d

Contents

mail_merge 2
mm_read_googledoc 4
mm_read_message 5
mm_read_message_googledoc 5
preview_mailmerge 6

Index **8**

`mail_merge` *Merges data into an email and send.*

Description

[Experimental]

Merges columns from a data frame into a markdown document using the `glue::glue_data()` function. The markdown can contain a yaml header for subject and cc line.

Note that only 'gmail' is supported at the moment, via gmailr::gm_send_message.

Before using `mail_merge()`, you must be authenticated to the gmail service. Use `gmailr::gm_auth()` to authenticate prior to starting the mail merge.

Usage

```
mail_merge(
  data,
  message,
  to_col = "email",
  send = c("preview", "draft", "immediately"),
  confirm = FALSE,
  sleep_preview = 1,
  sleep_send = 0.1
)
```

Arguments

`data`	A data frame or `tibble` with all the columns that should be glued into the message. Substitution is performed using `glue::glue_data()`‘
`message`	A list with components `yaml` and `body`. You can use `mm_read_message()` or `mm_read_message_googledoc()` to construct a message in this format.

to_col	The name of the column in data that contains the email address to send the message to.
send	A character string, one of: • "preview" : displays message in viewer without sending mail • "draft : writes message into "drafts" folder on gmail • "immediately" : sends email
confirm	If TRUE sends email without additional confirmation. If FALSE asks for confirmation before sending.
sleep_preview	If send == "preview" the number of seconds to sleep between each preview. See also preview_mailmerge
sleep_send	If send == "immediately" the number of seconds to sleep between each email send (to prevent gmail API 500 errors).

Value

Returns a list for every message, consisting of:

- msg: The message in mime format
- id: The gmailr response id
- type: preview, draft or sent
- success: TRUE if the message was sent successfully

See Also

preview_mailmerge

Examples

```
## ---- input-data -------------------------------------------------------

dat <-  data.frame(
  email      = c("friend@example.com", "foe@example.com"),
  first_name = c("friend", "foe"),
  thing      = c("something good", "something bad"),
  stringsAsFactors = FALSE
)

## ---- markdown-message -------------------------------------------------

msg <- '
---
subject: "**Hello, {first_name}**"
---

Hi, **{first_name}**

I am writing to tell you about **{thing}**.
```

```
{if (first_name == "friend") "Regards" else ""}


Me
'


## ---- mail-merge ----------------------------------------------------------

dat %>%
  mail_merge(msg)

if (interactive()) {
  dat %>%
    mail_merge(msg) %>%
    print()
}

## ---- display shiny gadget ------------------------------------------------

if (interactive()) {
  dat %>%
    mail_merge(msg) %>%
    preview_mailmerge()
}
```

`mm_read_googledoc` *Read a markdown file from google drive*

Description

Read a markdown file from google drive

Usage

```
mm_read_googledoc(id)
```

Arguments

`id`	Unique Google Drive identifier, passed to googledrive::drive_download

Value

the imported document

A list of character strings, containing the content of the google doc

mm_read_message *Parse a markdown document into yaml and body components.*

Description

You can use this to construct the message argument of mail_merge().

Usage

```
mm_read_message(txt)
```

Arguments

txt	A file in markdown format

Value

A list with components yaml and body

See Also

Other parsing functions: mm_read_message_googledoc()

mm_read_message_googledoc *Parse the email template inside a markdown document on google drive.*

Description

You can use this to construct the message argument of mail_merge().

Usage

```
mm_read_message_googledoc(id)
```

Arguments

id	Unique Google Drive identifier, passed to googledrive::drive_download

Value

A list of character strings

See Also

Other parsing functions: mm_read_message()

preview_mailmerge *Preview mailmerge as shiny gadget in RStudio preview pane.*

Description

[Experimental]

Usage

```
preview_mailmerge(x)
```

Arguments

x	mailmerge_preview object, from mail_merge

Value

A 'shiny' gadget, see also shiny::runGadget

See Also

mail_merge

Examples

```
## ---- input-data -------------------------------------------------------

dat <-  data.frame(
  email      = c("friend@example.com", "foe@example.com"),
  first_name = c("friend", "foe"),
  thing      = c("something good", "something bad"),
  stringsAsFactors = FALSE
)

## ---- markdown-message -------------------------------------------------

msg <- '
---
subject: "**Hello, {first_name}**"
---

Hi, **{first_name}**

I am writing to tell you about **{thing}**.

{if (first_name == "friend") "Regards" else ""}


Me
'
```

```
## ---- display shiny gadget ------------------------------------------------

if (interactive()) {
  dat %>%
    mail_merge(msg) %>%
    preview_mailmerge()
}
```

Index

∗ **parsing functions**
    mm_read_message, 5
    mm_read_message_googledoc, 5

glue::glue_data(), *2*
gmailr::gm_auth(), *2*
gmailr::gm_send_message, *2*
googledrive::drive_download, *4, 5*

mail_merge, 2, *6*
mail_merge(), *5*
mm_read_googledoc, 4
mm_read_message, 5, *5*
mm_read_message(), *2*
mm_read_message_googledoc, 5, *5*
mm_read_message_googledoc(), *2*

preview_mailmerge, *3*, 6

shiny::runGadget, *6*