

# Package ‘gmailr’

September 19, 2014

**Version** 0.5.0

**Date** 2014-7-25

**Maintainer** Jim Hester <james.f.hester@gmail.com>

**Author** Jim Hester

**Title** Access the Gmail RESTful API from R

**Description** An R interface to the Gmail RESTful API. Allows access to your Gmail messages, threads, drafts and labels from R.

**Depends** R (>= 3.0.0)

**Imports** httr, base64enc, magrittr, jsonlite

**Suggests** testthat, knitr

**License** MIT + file LICENSE

**URL** <https://github.com/jimhester/gmailr>

**BugReports** <https://github.com/jimhester/gmailr/issues>

**VignetteBuilder** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-09-19 00:34:52

## R topics documented:

as.character.mime . . . . .	2
attachment . . . . .	3
attach_part . . . . .	4
body . . . . .	4
create_draft . . . . .	5
create_label . . . . .	6

delete_label . . . . .	6
delete_message . . . . .	7
delete_thread . . . . .	7
draft . . . . .	8
drafts . . . . .	8
format . . . . .	9
gmailr . . . . .	10
gmail_auth . . . . .	10
history . . . . .	10
id . . . . .	11
import_message . . . . .	12
insert_message . . . . .	13
label . . . . .	13
labels . . . . .	14
message . . . . .	14
messages . . . . .	15
mime . . . . .	16
modify_message . . . . .	17
modify_thread . . . . .	17
print . . . . .	18
quoted_printable_encode . . . . .	19
save_attachment . . . . .	19
save_attachments . . . . .	20
send_draft . . . . .	21
send_message . . . . .	21
text_body . . . . .	22
thread . . . . .	22
threads . . . . .	23
to . . . . .	24
to.mime . . . . .	25
trash_message . . . . .	26
trash_thread . . . . .	26
untrash_message . . . . .	27
untrash_thread . . . . .	27
update_label . . . . .	28
%>% . . . . .	29

**Index** **30**

---

as.character.mime      *Convert a mime object to character representation*

---

**Description**

These functions convert a mime object into the final mime character representation.

**Usage**

```
## S3 method for class 'mime'  
as.character(x, ..., newline = "\r\n")
```

**Arguments**

x	object to convert
...	additional arguments ignored
newline	value to use as newline character

---

attachment	<i>Retrieve an attachment to a message</i>
------------	--

---

**Description**

Function to retrieve an attachment to a message by id of the attachment and message. To save the attachment use [save\\_attachment](#).

**Usage**

```
attachment(id, message_id, user_id = "me")
```

**Arguments**

id	id of the attachment
message_id	id of the parent message
user_id	gmail user_id to access, special value of 'me' indicates the authenticated user.

**References**

<https://developers.google.com/gmail/api/v1/reference/users/messages/attachments/get>

**Examples**

```
## Not run:  
my_attachment = attachment('a32e324b', '12345')  
save_attachment to a file  
save_attachment(my_attachment, 'photo.jpg')  
  
## End(Not run)
```

---

attach_part	<i>Attach an object to a mime message</i>
-------------	---

---

**Description**

Attach an object to a mime message

**Usage**

```
attach_part(mime, body, ...)
```

```
attach_file(mime, filename, type = guess_media(filename), ...)
```

**Arguments**

mime	object to attach to
body	data to attach
...	additional arguments put into the attr field of the object
filename	name of file to attach
type	mime type of the attached file

---

body	<i>Get the body text of a message or draft</i>
------	--

---

**Description**

Get the body text of a message or draft

**Usage**

```
body(x, ...)
```

```
## S3 method for class 'gmail_message'
body(x, collapse = FALSE, ...)
```

```
## S3 method for class 'gmail_draft'
body(x, ...)
```

**Arguments**

x	the object from which to retrieve the body
...	other parameters passed to methods
collapse	if 'FALSE' will return each formatted body in list, if 'TRUE' will collapse them together

**Examples**

```
## Not run:  
body(my_message)  
body(my_draft)  
  
## End(Not run)
```

---

create_draft	<i>Create a draft from a mime message</i>
--------------	---

---

**Description**

Create a draft from a mime message

**Usage**

```
create_draft(mail, user_id = "me", type = c("multipart", "media",  
      "resumable"))
```

**Arguments**

mail	mime mail message created by mime
user_id	gmail user_id to access, special value of 'me' indicates the authenticated user.
type	the type of upload to perform

**References**

<https://developers.google.com/gmail/api/v1/reference/users/drafts/create>

**Examples**

```
## Not run:  
create_draft(mime(from="you@me.com", to="any@one.com",  
      subject='hello', "how are you doing?"))  
  
## End(Not run)
```

---

create_label	<i>Create a new label</i>
--------------	---------------------------

---

**Description**

Function to create a label.

**Usage**

```
create_label(name, label_list_visibility = c("hide", "show", "show_unread"),  
            message_list_visibility = c("hide", "show"), user_id = "me")
```

**Arguments**

name	name to give to the new label
label_list_visibility	The visibility of the label in the label list in the Gmail web interface.
message_list_visibility	The visibility of messages with this label in the message list in the Gmail web interface.
user_id	gmail user_id to access, special value of 'me' indicates the authenticated user.

**References**

<https://developers.google.com/gmail/api/v1/reference/users/labels/create>

---

delete_label	<i>Permanently delete a label</i>
--------------	-----------------------------------

---

**Description**

Function to delete a label by id. This cannot be undone!

**Usage**

```
delete_label(id, user_id = "me")
```

**Arguments**

id	label id to retrieve
user_id	gmail user_id to access, special value of 'me' indicates the authenticated user.

**References**

<https://developers.google.com/gmail/api/v1/reference/users/labels/delete>

---

delete_message	<i>Permanently delete a single message</i>
----------------	--

---

**Description**

Function to delete a given message by id. This cannot be undone!

**Usage**

```
delete_message(id, user_id = "me")
```

**Arguments**

id	message id to access
user_id	gmail user_id to access, special value of 'me' indicates the authenticated user.

**References**

<https://developers.google.com/gmail/api/v1/reference/users/messages/delete>

**Examples**

```
## Not run:  
delete_message('12345')  
  
## End(Not run)
```

---

delete_thread	<i>Permanently delete a single thread.</i>
---------------	--

---

**Description**

Function to delete a given thread by id. This cannot be undone!

**Usage**

```
delete_thread(id, user_id = "me")
```

**Arguments**

id	thread id to access
user_id	gmail user_id to access, special value of 'me' indicates the authenticated user.

**References**

<https://developers.google.com/gmail/api/v1/reference/users/threads/delete>

**Examples**

```
## Not run:  
delete_thread(12345)  
  
## End(Not run)
```

---

draft	<i>Get a single draft</i>
-------	---------------------------

---

**Description**

Function to retrieve a given draft by id

**Usage**

```
draft(id, user_id = "me", format = c("full", "minimal", "raw"))
```

**Arguments**

id	draft id to access
user_id	gmail user_id to access, special value of 'me' indicates the authenticated user.
format	format of the draft returned

**References**

<https://developers.google.com/gmail/api/v1/reference/users/drafts/get>

**Examples**

```
## Not run:  
my_draft = draft('12345')  
  
## End(Not run)
```

---

drafts	<i>Get a list of drafts</i>
--------	-----------------------------

---

**Description**

Get a list of drafts possibly matching a given query string.

**Usage**

```
drafts(num_results = NULL, page_token = NULL, user_id = "me")
```



**Arguments**

num\_results      the number of results to return.  
page\_token        retrieve a specific page of results  
user\_id            gmail user\_id to access, special value of 'me' indicates the authenticated user.

**References**

<https://developers.google.com/gmail/api/v1/reference/users/drafts/list>

**Examples**

```
## Not run:  
my_drafts = drafts()  
  
first_10_drafts = drafts(10)  
  
## End(Not run)
```

---

format	<i>Format gmailr objects for pretty printing</i>
--------	--

---

**Description**

Format gmailr objects for pretty printing

**Usage**

```
## S3 method for class 'gmail_message'  
format(x, ...)  
  
## S3 method for class 'gmail_draft'  
format(x, ...)  
  
## S3 method for class 'gmail_messages'  
format(x, ...)  
  
## S3 method for class 'gmail_threads'  
format(x, ...)  
  
## S3 method for class 'gmail_drafts'  
format(x, ...)
```

**Arguments**

x                    object to format  
...                  additional arguments ignored

---

gmailr	<b>gmailr</b> <i>makes gmail access easy.</i>
--------	---

---

**Description**

gmailr provides an interface to the gmail api <https://developers.google.com/gmail/api/>

---

gmail_auth	<i>Setup oauth authentication for your gmail</i>
------------	--

---

**Description**

Setup oauth authentication for your gmail

**Usage**

```
gmail_auth(secret_file, scope = c("read_only", "modify", "compose", "full"))
```

**Arguments**

secret_file	the secret json file downloaded from <a href="https://cloud.google.com/console#/project">https://cloud.google.com/console#/project</a>
scope	the authentication scope to use

**Examples**

```
## Not run:  
gmail_auth("file", "compose")  
  
## End(Not run)
```

---

history	<i>Retrieve change history for the inbox</i>
---------	--

---

**Description**

Retrieves the history results in chronological order

**Usage**

```
history(start_history_id = NULL, num_results = NULL, label_id = NULL,  
        page_token = NULL, user_id = "me")
```

**Arguments**

<code>start_history_id</code>	the point to start the history. The historyId can be obtained from a message, thread or previous list response.
<code>num_results</code>	the number of results to return, max per page is 100
<code>label_id</code>	filter history only for this label
<code>page_token</code>	retrieve a specific page of results
<code>user_id</code>	gmail user_id to access, special value of 'me' indicates the authenticated user.

**References**

<https://developers.google.com/gmail/api/v1/reference/users/history/list>

**Examples**

```
## Not run:
my_history = history("10")

## End(Not run)
```

---

id	<i>Get the id of a gmailr object</i>
----	--------------------------------------

---

**Description**

Get the id of a gmailr object

**Usage**

```
id(x, ...)

## S3 method for class 'gmail_message'
id(x, ...)

## S3 method for class 'gmail_draft'
id(x, ...)

## S3 method for class 'gmail_messages'
id(x, what = c("message_id", "thread_id"), ...)

## S3 method for class 'gmail_drafts'
id(x, what = c("draft_id", "message_id", "thread_id"),
    ...)

## S3 method for class 'gmail_threads'
id(x, ...)
```

**Arguments**

x	the object from which to retrieve the id
...	other parameters passed to methods
what	the type of id to return

**Examples**

```
## Not run:
id(my_message)
id(my_draft)

## End(Not run)
```

---

import_message	<i>Import a message into the gmail mailbox from a mime message</i>
----------------	--

---

**Description**

Import a message into the gmail mailbox from a mime message

**Usage**

```
import_message(mail, user_id = "me", label_ids = NULL,
               type = c("multipart", "media", "resumable"),
               internal_date_source = c("dateHeader", "recievedTime"))
```

**Arguments**

mail	mime mail message created by mime
user_id	gmail user_id to access, special value of 'me' indicates the authenticated user.
label_ids	optional label ids to apply to the message
type	the type of upload to perform
internal_date_source	whether to date the object based on the date of the message or when it was received by gmail.

**References**

<https://developers.google.com/gmail/api/v1/reference/users/messages/import>

**Examples**

```
## Not run:
import_message(mime(from="you@me.com", to="any@one.com",
                   subject='hello', "how are you doing?"))

## End(Not run)
```

---

insert_message	<i>Insert a message into the gmail mailbox from a mime message</i>
----------------	--

---

**Description**

Insert a message into the gmail mailbox from a mime message

**Usage**

```
insert_message(mail, user_id = "me", label_ids = NULL,
              type = c("multipart", "media", "resumable"),
              internal_date_source = c("dateHeader", "recievedTime"))
```

**Arguments**

mail	mime mail message created by mime
user_id	gmail user_id to access, special value of 'me' indicates the authenticated user.
label_ids	optional label ids to apply to the message
type	the type of upload to perform
internal_date_source	whether to date the object based on the date of the message or when it was received by gmail.

**References**

<https://developers.google.com/gmail/api/v1/reference/users/messages/insert>

**Examples**

```
## Not run:
insert_message(mime(from="you@me.com", to="any@one.com",
                  subject='hello', "how are you doing?"))

## End(Not run)
```

---

label	<i>Get a specific label</i>
-------	-----------------------------

---

**Description**

Get a specific label by id and user\_id.

**Usage**

```
label(id, user_id = "me")
```

**Arguments**

id                    label id to retrieve  
 user\_id              gmail user\_id to access, special value of 'me' indicates the authenticated user.

**References**

<https://developers.google.com/gmail/api/v1/reference/users/labels/get>

---

labels	<i>Get a list of all labels</i>
--------	---------------------------------

---

**Description**

Get a list of all labels for a user.

**Usage**

```
labels(user_id = "me")
```

**Arguments**

user\_id              gmail user\_id to access, special value of 'me' indicates the authenticated user.

**References**

<https://developers.google.com/gmail/api/v1/reference/users/labels/list>

**Examples**

```
## Not run:
my_labels = labels()

## End(Not run)
```

---

message	<i>Get a single message</i>
---------	-----------------------------

---

**Description**

Function to retrieve a given message by id

**Usage**

```
message(id, user_id = "me", format = c("full", "minimal", "raw"))
```

**Arguments**

id	message id to access
user_id	gmail user_id to access, special value of 'me' indicates the authenticated user.
format	format of the message returned

**References**

<https://developers.google.com/gmail/api/v1/reference/users/messages>

**Examples**

```
## Not run:
my_message = message(12345)

## End(Not run)
```

---

messages	<i>Get a list of messages</i>
----------	-------------------------------

---

**Description**

Get a list of messages possibly matching a given query string.

**Usage**

```
messages(search = NULL, num_results = NULL, label_ids = NULL,
         include_spam_trash = NULL, user_id = "me", page_token = NULL)
```

**Arguments**

search	query to use, same format as gmail search box.
num_results	the number of results to return.
label_ids	restrict search to given labels
include_spam_trash	boolean whether to include the spam and trash folders in the search
user_id	gmail user_id to access, special value of 'me' indicates the authenticated user.
page_token	retrieve a specific page of results

**References**

<https://developers.google.com/gmail/api/v1/reference/users/messages/list>

**Examples**

```
## Not run:
#Search for R, return 10 results using label 1 including spam and trash folders
my_messages = messages("R", 10, "label_1", TRUE)

## End(Not run)
```

---

 mime

---

*Create a mime formatted message object*


---

**Description**

Create a mime formatted message object

**Usage**

```
mime(..., attr = NULL, body = NULL, parts = list())
```

**Arguments**

...	You can set header values initially, or use
attr	attributes to pass to the model
body	text to use for the body
parts	mime parts to pass to the model <a href="#">common_fields</a> set functions to set them after object creation.

**See Also**

[common\\_fields](#), [body](#)

**Examples**

```
# using the field functions
msg = mime() %>%
  from("james.f.hester@gmail.com") %>%
  to("CRAN@R-project.org") %>%
  text_body("Please don't reject my package")

# alternatively you can set the fields using mime, however you hav()e
# to use properly formatted MIME names
msg = mime(From="james.f.hester@gmail.com",
           To="CRAN@R-project.org") %>%
  html_body("<b>Please<\b> don't reject my package")
```



---

modify_message	<i>Modify the labels on a message</i>
----------------	---------------------------------------

---

**Description**

Function to modify the labels on a given message by id.

**Usage**

```
modify_message(id, add_labels = character(0), remove_labels = character(0),  
              user_id = "me")
```

**Arguments**

id	message id to access
add_labels	labels to add to the specified message
remove_labels	labels to remove from the specified message
user_id	gmail user_id to access, special value of 'me' indicates the authenticated user.

**References**

<https://developers.google.com/gmail/api/v1/reference/users/messages/modify>

**Examples**

```
## Not run:  
modify_message(12345, add_labels='label_1')  
modify_message(12345, remove_labels='label_1')  
#add and remove at the same time  
modify_message(12345, add_labels='label_2', remove_labels='label_1')  
  
## End(Not run)
```

---

modify_thread	<i>Modify the labels on a thread</i>
---------------	--------------------------------------

---

**Description**

Function to modify the labels on a given thread by id.

**Usage**

```
modify_thread(id, add_labels = character(0), remove_labels = character(0),  
             user_id = "me")
```

**Arguments**

id thread id to access  
 add\_labels labels to add to the specified thread  
 remove\_labels labels to remove from the specified thread  
 user\_id gmail user\_id to access, special value of 'me' indicates the authenticated user.

**References**

<https://developers.google.com/gmail/api/v1/reference/users/threads/modify>

**Examples**

```

## Not run:
modify_thread(12345, add_labels='label_1')
modify_thread(12345, remove_labels='label_1')
#add and remove at the same time
modify_thread(12345, add_labels='label_2', remove_labels='label_1')

## End(Not run)

```

---

print

*Print gmailr objects*

---

**Description**

Print gmailr objects

**Usage**

```

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

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

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

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

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

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

```

**Arguments**

x	object to print
...	additional arguments ignored

---

`quoted_printable_encode`*Encode text using quoted printable*

---

**Description**

Does not do any line wrapping of the output to 76 characters. Implementation derived from the perl MIME::QuotedPrint

**Usage**`quoted_printable_encode(data)`**Arguments**

data	data to encode
------	----------------

**References**

<http://search.cpan.org/~gaas/MIME-Base64-3.14/QuotedPrint.pm>

---

`save_attachment`*save the attachment to a file*

---

**Description**

this only works on attachments retrieved with [attachment](#). To save an attachment directly from a message see [save\\_attachments](#)

**Usage**`save_attachment(x, filename)`**Arguments**

x	attachment to save
filename	location to save to

### Examples

```
## Not run:
my_attachment = attachment('a32e324b', '12345')
save_attachment to a file
save_attachment(my_attachment, 'photo.jpg')

## End(Not run)
```

---

save_attachments	<i>Save attachments to a message</i>
------------------	--------------------------------------

---

### Description

Function to retrieve and save all of the attachments to a message by id of the message.

### Usage

```
save_attachments(x, attachment_id = NULL, path = "", user_id = "me")
```

### Arguments

x	message with attachment
attachment_id	id of the attachment to save, if none specified saves all attachments
path	where to save the attachments
user_id	gmail user_id to access, special value of 'me' indicates the authenticated user.

### References

<https://developers.google.com/gmail/api/v1/reference/users/messages/attachments/get>

### Examples

```
## Not run:
save all attachments
save_attachments(my_message)
save a specific attachment
save_attachments(my_message, 'a32e324b')

## End(Not run)
```

---

send_draft	<i>Send a draft</i>
------------	---------------------

---

**Description**

Send a draft to the recipients in the To, CC, and Bcc headers.

**Usage**

```
send_draft(id, upload_type = c("media", "multipart", "resumable"),
           user_id = "me")
```

**Arguments**

id	the draft id to send
upload_type	type of upload request
user_id	gmail user_id to access, special value of 'me' indicates the authenticated user.

**References**

<https://developers.google.com/gmail/api/v1/reference/users/drafts/send>

**Examples**

```
## Not run:
send_draft(12345)

## End(Not run)
```

---

send_message	<i>Send a message from a mime message</i>
--------------	---

---

**Description**

Send a message from a mime message

**Usage**

```
send_message(mail, user_id = "me", label_ids = NULL, type = c("multipart",
                    "media", "resumable"))
```

**Arguments**

mail	mime mail message created by mime
user_id	gmail user_id to access, special value of 'me' indicates the authenticated user.
label_ids	optional label ids to apply to the message
type	the type of upload to perform

**References**

<https://developers.google.com/gmail/api/v1/reference/users/messages/send>

**Examples**

```
## Not run:
send_message(mime(from="you@me.com", to="any@one.com",
                  subject='hello', "how are you doing?"))

## End(Not run)
```

---

text_body	<i>Add a text or html body to a mime message</i>
-----------	--

---

**Description**

if called without a body returns the current body

**Usage**

```
text_body(mime, body, ...)
```

```
html_body(mime, body, ...)
```

**Arguments**

mime	message to add the body to
body	the body to add to the message
...	additional parameters to put in the attr field

---

thread	<i>Get a single thread</i>
--------	----------------------------

---

**Description**

Function to retrieve a given thread by id

**Usage**

```
thread(id, user_id = "me")
```

**Arguments**

id	thread id to access
user_id	gmail user_id to access, special value of 'me' indicates the authenticated user.

## References

<https://developers.google.com/gmail/api/v1/reference/users/thread>

## Examples

```
## Not run:  
my_thread = thread(12345)  
  
## End(Not run)
```

---

threads

*Get a list of threads*

---

## Description

Get a list of threads possibly matching a given query string.

## Usage

```
threads(search = NULL, num_results = NULL, page_token = NULL,  
        label_ids = NULL, include_spam_trash = NULL, user_id = "me")
```

## Arguments

search	query to use, same format as gmail search box.
num_results	the number of results to return.
page_token	retrieve a specific page of results
label_ids	restrict search to given labels
include_spam_trash	boolean whether to include the spam and trash folders in the search
user_id	gmail user_id to access, special value of 'me' indicates the authenticated user.

## References

<https://developers.google.com/gmail/api/v1/reference/users/threads/list>

## Examples

```
## Not run:  
my_threads = threads()  
  
first_10_threads = threads(10)  
  
## End(Not run)
```

---

to *Methods to get values from message or drafts*

---

## Description

Methods to get values from message or drafts

## Usage

```
to(x, ...)  
  
## S3 method for class 'gmail_message'  
to(x, ...)  
  
## S3 method for class 'gmail_draft'  
to(x, ...)  
  
from(x, ...)  
  
## S3 method for class 'gmail_message'  
from(x, ...)  
  
cc.gmail_message(x, ...)  
  
cc.gmail_draft(x, ...)  
  
bcc.gmail_message(x, ...)  
  
bcc.gmail_draft(x, ...)  
  
date(x, ...)  
  
## S3 method for class 'gmail_message'  
date(x, ...)  
  
## S3 method for class 'gmail_draft'  
date(x, ...)  
  
subject(x, ...)  
  
## S3 method for class 'gmail_message'  
subject(x, ...)  
  
## S3 method for class 'gmail_draft'  
subject(x, ...)
```



**Arguments**

x                    the object from which to get or set the field  
...                   other parameters passed to methods

**See Also**

[common\\_fields](#)

---

to.mime                    *Accessor functions for mime messages*

---

**Description**

Accessor functions for mime messages

**Usage**

```
## S3 method for class 'mime'  
to(x, vals, ...)  
  
## S3 method for class 'mime'  
from(x, val, ...)  
  
cc(x, vals, ...)  
  
bcc(x, vals, ...)  
  
## S3 method for class 'mime'  
subject(x, val, ...)
```

**Arguments**

x                    the object whose fields you are setting  
vals                  one or more values to use, will be joined by commas  
...                   other arguments ignored  
val                   the value to set

---

trash_message	<i>Send a single message to the trash</i>
---------------	---

---

**Description**

Function to trash a given message by id. This can be undone by [untrash\\_message](#).

**Usage**

```
trash_message(id, user_id = "me")
```

**Arguments**

id	message id to access
user_id	gmail user_id to access, special value of 'me' indicates the authenticated user.

**References**

<https://developers.google.com/gmail/api/v1/reference/users/messages/trash>

**Examples**

```
## Not run:  
trash_message('12345')  
  
## End(Not run)
```

---

trash_thread	<i>Send a single thread to the trash</i>
--------------	--

---

**Description**

Function to trash a given thread by id. This can be undone by [untrash\\_thread](#).

**Usage**

```
trash_thread(id, user_id = "me")
```

**Arguments**

id	thread id to access
user_id	gmail user_id to access, special value of 'me' indicates the authenticated user.

**References**

<https://developers.google.com/gmail/api/v1/reference/users/threads/trash>

**Examples**

```
## Not run:  
trash_thread(12345)  
  
## End(Not run)
```

---

untrash_message	<i>Remove a single message from the trash</i>
-----------------	---

---

**Description**

Function to trash a given message by id. This can be undone by [untrash\\_message](#).

**Usage**

```
untrash_message(id, user_id = "me")
```

**Arguments**

id	message id to access
user_id	gmail user_id to access, special value of 'me' indicates the authenticated user.

**References**

<https://developers.google.com/gmail/api/v1/reference/users/messages/trash>

**Examples**

```
## Not run:  
untrash_message('12345')  
  
## End(Not run)
```

---

untrash_thread	<i>Remove a single thread from the trash.</i>
----------------	---

---

**Description**

Function to untrash a given thread by id. This can reverse the results of a previous [trash\\_thread](#).

**Usage**

```
untrash_thread(id, user_id = "me")
```

**Arguments**

id                    thread id to access  
 user\_id              gmail user\_id to access, special value of 'me' indicates the authenticated user.

**References**

<https://developers.google.com/gmail/api/v1/reference/users/threads/untrash>

**Examples**

```
## Not run:
untrash_thread(12345)

## End(Not run)
```

---

update_label	<i>Update a existing label.</i>
--------------	---------------------------------

---

**Description**

Get a specific label by id and user\_id. update\_label\_patch is identical to update\_label but the latter uses **HTTP PATCH** to allow partial update.

**Usage**

```
update_label(id, label, user_id = "me")
update_label_patch(id, label, user_id = "me")
```

**Arguments**

id                    label id to update  
 label                the label fields to update  
 user\_id              gmail user\_id to access, special value of 'me' indicates the authenticated user.

**References**

<https://developers.google.com/gmail/api/v1/reference/users/labels/update>

<https://developers.google.com/gmail/api/v1/reference/users/labels/patch>

---

`%>%`*Pipe statements*

---

**Description**

Like dplyr and ggvis gmailr also uses the pipe function, `%>%` to turn function composition into a series of imperative statements.

**Arguments**

`lhs, rhs`      A visualisation and a function to apply to it

**Examples**

```
# Instead of  
to(mime(), 'someone@somewhere.com')  
# you can write  
mime() %>% to('someone@somewhere.com')
```

# Index

[%>%](#), [29](#)

[as.character.mime](#), [2](#)  
[attach\\_file\(attach\\_part\)](#), [4](#)  
[attach\\_part](#), [4](#)  
[attachment](#), [3](#), [19](#)

[bcc\(to.mime\)](#), [25](#)  
[bcc.gmail\\_draft\(to\)](#), [24](#)  
[bcc.gmail\\_message\(to\)](#), [24](#)  
[body](#), [4](#), [16](#)

[cc\(to.mime\)](#), [25](#)  
[cc.gmail\\_draft\(to\)](#), [24](#)  
[cc.gmail\\_message\(to\)](#), [24](#)  
[common\\_fields](#), [16](#), [25](#)  
[common\\_fields\(to.mime\)](#), [25](#)  
[create\\_draft](#), [5](#)  
[create\\_label](#), [6](#)

[date\(to\)](#), [24](#)  
[delete\\_label](#), [6](#)  
[delete\\_message](#), [7](#)  
[delete\\_thread](#), [7](#)  
[draft](#), [8](#)  
[drafts](#), [8](#)

[format](#), [9](#)  
[from\(to\)](#), [24](#)  
[from.mime\(to.mime\)](#), [25](#)

[gmail\\_auth](#), [10](#)  
[gmailr](#), [10](#)  
[gmailr-package\(gmailr\)](#), [10](#)

[history](#), [10](#)  
[html\\_body\(text\\_body\)](#), [22](#)

[id](#), [11](#)  
[import\\_message](#), [12](#)  
[insert\\_message](#), [13](#)

[label](#), [13](#)  
[labels](#), [14](#)

[message](#), [14](#)  
[messages](#), [15](#)  
[mime](#), [16](#)  
[modify\\_message](#), [17](#)  
[modify\\_thread](#), [17](#)

[print](#), [18](#)

[quoted\\_printable\\_encode](#), [19](#)

[save\\_attachment](#), [3](#), [19](#)  
[save\\_attachments](#), [19](#), [20](#)  
[send\\_draft](#), [21](#)  
[send\\_message](#), [21](#)  
[subject\(to\)](#), [24](#)  
[subject.mime\(to.mime\)](#), [25](#)

[text\\_body](#), [22](#)  
[thread](#), [22](#)  
[threads](#), [23](#)  
[to](#), [24](#)  
[to.mime](#), [25](#)  
[trash\\_message](#), [26](#)  
[trash\\_thread](#), [26](#), [27](#)

[untrash\\_message](#), [26](#), [27](#), [27](#)  
[untrash\\_thread](#), [26](#), [27](#)  
[update\\_label](#), [28](#)  
[update\\_label\\_patch\(update\\_label\)](#), [28](#)