Package ‘gmailr’

August 29, 2016

Version 0.7.1

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

Author Jim Hester

Title Access the Gmail RESTful API

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

Depends R (>= 3.0.0)

Imports httr, base64enc, magrittr, jsonlite, mime, crayon

Suggests testthat, knitr, methods, rmarkdown

License MIT + file LICENSE

URL https://github.com/jimhester/gmailr

BugReports https://github.com/jimhester/gmailr/issues

VignetteBuilder knitr

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-04-12 01:02:42

R topics documented:

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

Description

This function converts a mime object into a character vector

Usage

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

Arguments

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

`attachment` *Retrieve an attachment to a message*

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

See Also

Other message: `delete_message`, `import_message`, `insert_message`, `messages`, `message`, `modify_message`, `save_attachments`, `save_attachment`, `send_message`, `trash_message`, `untrash_message`

Examples

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

## End(Not run)
```

`body` *Get the body text of a message or draft*

Description

Get the body text of a message or draft

Usage

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

Arguments

`x`	the object from which to retrieve the body
`...`	other parameters passed to methods

Examples

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

## End(Not run)
```

`clear_token` *Clear the current oauth token*

Description

Clear the current oauth token

Usage

```
clear_token()
```

`create_draft` *Create a draft from a mime message*

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` *Create a new label*

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

See Also

Other label: `delete_label`, `labels`, `label`, `update_label`

`delete_label` *Permanently delete a label*

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

See Also

Other label: `create_label`, `labels`, `label`, `update_label`

`delete_message` *Permanently delete a single message*

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

See Also

Other message: `attachment`, `import_message`, `insert_message`, `messages`, `message`, `modify_message`, `save_attachments`, `save_attachment`, `send_message`, `trash_message`, `untrash_message`

Examples

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

## End(Not run)
```

`delete_thread` *Permanently delete a single thread.*

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

See Also

Other thread: modify_thread, threads, thread, trash_thread, untrash_thread

Examples

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

## End(Not run)
```

---

draft *Get a single draft*

Description

Function to retrieve a given draft by <-

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

See Also

Other draft: drafts, send_draft

Examples

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

## End(Not run)
```

drafts *Get a list of drafts*

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

See Also

Other draft: `draft`, `send_draft`

Examples

```
## Not run:
my_drafts = drafts()

first_10_drafts = drafts(10)

## End(Not run)
```

gmailr **gmailr** *makes gmail access easy.*

Description

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

gmail_auth *Setup oauth authentication for your gmail*

Description

Setup oauth authentication for your gmail

Usage

```
gmail_auth(scope = c("read_only", "modify", "compose", "full"), id = the$id,
  secret = the$secret, secret_file = NULL)
```

Arguments

scope	the authentication scope to use
id	the client_id to use for authentication
secret	the client secret to use for authentication
secret_file	the secret json file downloaded from https://console.cloud.google.com

See Also

use_secret_file to set the default id and secret to a different value than the default.

Examples

```
## Not run:
gmail_auth("compose")

## End(Not run)
```

history *Retrieve change history for the inbox*

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

`start_history_id`	the point to start the history. The historyId can be obtained from a message, thread or previous list response.
`num_results`	the number of results to return, max per page is 100
`label_id`	filter history only for this label
`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/history/list

Examples

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

## End(Not run)
```

id *Get the id of a gmailr object*

Description

Get the id of a gmailr object

Usage

```
id(x, ...)

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

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 *Import a message into the gmail mailbox from a mime message*

Description

Import a message into the gmail mailbox from a mime message

Usage

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

Arguments

`mail`	mime mail message created by mime
`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.
`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/import

See Also

Other message: `attachment`, `delete_message`, `insert_message`, `messages`, `message`, `modify_message`, `save_attachments`, `save_attachment`, `send_message`, `trash_message`, `untrash_message`

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 *Insert a message into the gmail mailbox from a mime message*

Description

Insert a message into the gmail mailbox from a mime message

Usage

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

Arguments

mail	mime mail message created by mime
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.
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/insert

See Also

Other message: attachment, delete_message, import_message, messages, message, modify_message, save_attachments, save_attachment, send_message, trash_message, untrash_message

Examples

```
## Not run:
insert_message(mime(From="you@me.com", To="any@one.com",
                          Subject="hello", "how are you doing?"))

## End(Not run)
```

`label` *Get a specific label*

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

See Also

Other label: `create_label`, `delete_label`, `labels`, `update_label`

`labels` *Get a list of all labels*

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

See Also

Other label: `create_label`, `delete_label`, `label`, `update_label`

Examples

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

## End(Not run)
```

last_response *Response from the last query*

Description

Response from the last query

Usage

```
last_response()
```

message *Get a single message*

Description

Function to retrieve a given message by id

Usage

```
message(id, user_id = "me", format = c("full", "metadata", "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

See Also

Other message: attachment, delete_message, import_message, insert_message, messages, modify_message, save_attachments, save_attachment, send_message, trash_message, untrash_message

Examples

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

## End(Not run)
```

---

messages *Get a list of messages*

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, page_token = NULL, user_id = "me")
```

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
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/messages/list

See Also

Other message: attachment, delete_message, import_message, insert_message, message, modify_message, save_attachments, save_attachment, send_message, trash_message, untrash_message

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

These functions create a MIME message. They can be created atomically using ‘mime()‘ or iteratively using the various accessors.

Usage

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

## S3 method for class 'mime'
to(x, vals, ...)

## S3 method for class 'mime'
from(x, val, ...)

## S3 method for class 'mime'
cc(x, vals, ...)

## S3 method for class 'mime'
bcc(x, vals, ...)

## S3 method for class 'mime'
subject(x, val, ...)

text_body(mime, body, ...)

html_body(mime, body, ...)

attach_part(mime, part, ...)

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

Arguments

`...`	additional parameters to put in the attr field
`attr`	attributes to pass to the message
`body`	Message body.
`parts`	mime parts to pass to the message
`x`	the object whose fields you are setting
`vals`	one or more values to use, will be joined by commas
`val`	the value to set
`mime`	message.

part	Message part to attach
filename	name of file to attach
type	mime type of the attached file

Examples

```
# using the field functions
msg = mime() %>%
 from("james.f.hester@gmail.com") %>%
 to("asdf@asdf.com") %>%
 text_body("Test Message")

# alternatively you can set the fields using mime(), however you have
#  to use properly formatted MIME names
msg = mime(From="james.f.hester@gmail.com",
                   To="asdf@asdf.com") %>%
        html_body("<b>Test<\b> Message")
```

---

modify_message *Modify the labels on a message*

Description

Function to modify the labels on a given message by id. Note you need to use the label ID as arguments to this function, not the label name.

Usage

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

Arguments

id	message id to access
add_labels	label IDs to add to the specified message
remove_labels	label IDs 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

See Also

Other message: attachment, delete_message, import_message, insert_message, messages, message, save_attachments, save_attachment, send_message, trash_message, untrash_message

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 *Modify the labels on a thread*

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

See Also

Other thread: `delete_thread`, `threads`, `thread`, `trash_thread`, `untrash_thread`

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)
```

quoted_printable_encode *Encode text using quoted printable*

Description

Does no 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

See Also

Other message: `attachment`, `delete_message`, `import_message`, `insert_message`, `messages`, `message`, `modify_message`, `save_attachments`, `send_message`, `trash_message`, `untrash_message`

Examples

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

## End(Not run)
```

---

save_attachments *Save attachments to a message*

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

See Also

Other message: attachment, delete_message, import_message, insert_message, messages, message, modify_message, save_attachment, send_message, trash_message, untrash_message

Examples

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

## End(Not run)
```

send_draft *Send a draft*

Description

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

Usage

```
send_draft(draft, user_id = "me")
```

Arguments

draft	the draft to send
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

See Also

Other draft: drafts, draft

Examples

```
## Not run:
draft <- create_draft(mime(From="you@me.com", To="any@one.com",
                      Subject="hello", "how are you doing?"))
send_draft(draft)

## End(Not run)
```

send_message *Send a message from a mime message*

Description

Send a message from a mime message

Usage

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

Arguments

`mail`	mime mail message created by mime
`type`	the type of upload to perform
`thread_id`	the id of the thread to send from.
`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/send

See Also

Other message: `attachment`, `delete_message`, `import_message`, `insert_message`, `messages`, `message`, `modify_message`, `save_attachments`, `save_attachment`, `trash_message`, `untrash_message`

Examples

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

## End(Not run)
```

`thread` *Get a single thread*

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/threads

See Also

Other thread: `delete_thread`, `modify_thread`, `threads`, `trash_thread`, `untrash_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

See Also

Other thread: delete_thread, modify_thread, thread, trash_thread, untrash_thread

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, ...)

from(x, ...)

cc(x, ...)

bcc(x, ...)

date(x, ...)

subject(x, ...)
```

Arguments

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

trash_message *Send a single message to the trash*

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

See Also

Other message: attachment, delete_message, import_message, insert_message, messages, message, modify_message, save_attachments, save_attachment, send_message, untrash_message

Examples

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

## End(Not run)
```

trash_thread *Send a single thread to the trash*

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

See Also

Other thread: delete_thread, modify_thread, threads, thread, untrash_thread

Examples

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

## End(Not run)
```

untrash_message *Remove a single message from the trash*

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

See Also

Other message: attachment, delete_message, import_message, insert_message, messages, message, modify_message, save_attachments, save_attachment, send_message, trash_message

Examples

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

## End(Not run)
```

untrash_thread *Remove a single thread from the trash.*

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

See Also

Other thread: delete_thread, modify_thread, threads, thread, trash_thread

Examples

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

## End(Not run)
```

---

update_label *Update a existing label.*

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

See Also

Other label: create_label, delete_label, labels, label

Other label: create_label, delete_label, labels, label

`use_secret_file` *Use information from a secret file*

Description

This function sets the default secret and client_id to those in the secret file

Usage

```
use_secret_file(filename)
```

Arguments

`filename`	the filename of the file

Index