Package ‘mRpostman’

April 20, 2020

Type Package

Title IMAP Toolkit

Version 0.3.1

Date 2020-04-18

Description IMAP4rev1 (Internet Message Access Protocol, Version 4rev1) functionalities based on the RFC 3501 manual (Crispin, 2003, <doi:10.17487/RFC3501>), its updates, and other related documents. 'mRpostman' makes extensive use of 'curl' and 'libcurl' capabilities, providing functions for mailboxes and electronic messages manipulation.

License GPL-3

Encoding UTF-8

LazyData true

Imports curl, stringr, magrittr, assertthat, base64enc, utils

Depends R (>= 3.1.0)

BugReports `https://github.com/allanvc/mRpostman/issues`

SystemRequirements libcurl: libcurl-devel (rpm) or libcurl4-openssl-dev (deb)

Suggests knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 6.1.1

NeedsCompilation no

Author Allan Quadros [aut, cre]

Maintainer Allan Quadros <`allanvcq@gmail.com`>

Repository CRAN

Date/Publication 2020-04-20 14:30:02 UTC

R topics documented:

`mRpostman-package` *IMAP Tools for R*

Description

mRpostman provides multiple IMAP (Internet Message Access Protocol) commands based on the RFC 3501 manual (Crispin, 2003), its updates, and other related documents. The goal is to make it easy to connect to your IMAP server and execute commands such as listing mailboxes, searching and fetching messages in a tidy way.

Author(s)

Author & Mantainer: Allan Quadros <`allanvcq@gmail.com`>

References

Crispin, M., *INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1*, RFC 3501, DOI: 10.17487/RFC3501, March 2003, `https://www.rfc-editor.org/info/rfc3501`.

Ooms, J. *curl: A Modern and Flexible Web Client for R*. R package version 3.3, 2019, `https://CRAN.R-project.org/package=curl`.

Stenberg, D. *Libcurl - The Multiprotocol File Transfer Library*, `https://curl.haxx.se/libcurl/`.

See Also

Useful links:

- `mRpostman official website:` `https://allanvc.github.io/mRpostman`

add_flags *Add Flag(s) To Messages*

Description

Add one or more flags to one or more messages.

Usage

```
add_flags(imapconf, msg_id, by = "MSN", flags_to_set, retries = 2)
```

Arguments

`imapconf`	An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.
`msg_id`	A numeric vetor containing one or more messages ids.
`by`	Which id (`MSN` or `UID`) to use when searching for messages. `"MSN"` or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. `"UID"` or unique identifier is always the same during the life cycle of a message. Default is `"MSN"`.
`flags_to_set`	A character vector containing one ore more flag names to add to or replace in the specified message ids.
`retries`	Number of attempts to connect and execute the command. Default is `2`.

Value

An (invisible) `list` of length `2` containing the `imapconf` object and the previously inputed message ids (parameter `msg_id`).

Note

add_flags, remove_flags and replace_flags, accepts not only flags but also keywords (such as `$Phishing` on Gmail's IMAP server) to add, remove or replace.

IMAP servers do not allow setting the negative version of a flag when a message already has the positive version of it. If a message with `MSN 1` already has the "SEEN" flag, it is not allowed to add "UNSEEN" to that. Instead, you have to first remove the "SEEN" flag `remove_flags(imapconf,msg_id = 1,"SEEN")` and only then do `add_flags(imapconf,msg_id = 1,"SEEN")`. Another option is to complete override all the flags of a message or of a set of messages using `replace_flags(imapconf,msg_id = 1,"SEEN")`.

See Also

Other miscellaneous: `copy_msg`, `count_msgs`, `delete_msg`, `expunge`, `get_max_id`, `get_min_id`, `move_msg`, `remove_flags`, `replace_flags`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

results <- imapconf %>%
    select_mailbox(mbox = "TAM") %>%
    search_before(date_char = "10-may-2012", by = "UID") %$% #exposition pipe - two argg
    add_flags(imapconf = imapconf, msg_id = msg_id, flags_to_set = "RECENT")


## End(Not run)
```

AND *Relational Operators - Helper Functions for Custom Search*

Description

Allows a combination of criteria helper functions such as such as before, since, on, sent_before, sent_since, sent_on, flag, string, smaller_than, larger_than, younger_than, or younger_than, in order to execute a multiple criteria custom search.

Usage

```
AND(..., negate = FALSE)
```

Arguments

`...`	a combination of criteria helper functions with its arguments.
`negate`	If `TRUE`, negates the search and seeks for "NOT search_criterion". Default is `FALSE`.

Value

A search string to be used as a `custom_request` parameter in custom_search function.

See Also

Other customsearch helper functions: `OR`, `before`, `flag`, `larger_than`, `older_than`, `on`, `sent_before`, `sent_on`, `sent_since`, `since`, `smaller_than`, `string`, `younger_than`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search
result <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    custom_search(custom_request = AND(sent_since(date.char = "17-Apr-2019"),
                                       smaller_than(size = 512000),
                                       negate = TRUE))
# searches for messages NOT SentSince "17-Apr-2019" AND NOT Smaller Than 512KB.


## End(Not run)
```

before *Criteria Helper Functions for Custom Search*

Description

Criteria helper functions to be combined inside a operator helper function as a custom request in custom_search.

Usage

```
before(date_char, negate = FALSE)
```

Arguments

`date_char`	A character vector with format "DD-Mon-YYYY", e.g. "01-Apr-2019". We opted not to use objects of type "date", since IMAP servers like this not so common date format.
`negate`	If `TRUE`, negates the search and seeks for "NOT search_criterion". Default is `FALSE`.

Value

A search string to be used as a `custom_request` parameter in custom_search function.

See Also

Other customsearch helper functions: `AND`, `OR`, `flag`, `larger_than`, `older_than`, `on`, `sent_before`, `sent_on`, `sent_since`, `since`, `smaller_than`, `string`, `younger_than`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search
result <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    custom_search(custom_request = AND(before(date_char = "17-Apr-2019"),
                                       smaller_than(size = 512000),
                                       negate = TRUE))
# searches for messages NOT Before "17-Apr-2019" AND NOT Smaller Than  512KB.


## End(Not run)
```

`configure_imap` *IMAP Settings*

Description

Configure IMAP server settings.

Usage

```
configure_imap(url, username, password, show_pass = FALSE,
  use_ssl = TRUE, verbose = FALSE, buffersize = 16000,
  fresh_connect = FALSE, timeout_ms = 5000, ...)
```

Arguments

`url` String containing the IMAP server address

`username` String containing the username.

`password` String containing user's password.

`show_pass` How to store user's password in imapconf object. Default is `FALSE`.

`use_ssl` A Logical indicating the use or not of Secure Sockets Layer encryption when connecting to IMAP server. Default is `TRUE`.

`verbose` If `FALSE`, mutes the output from the server. Default is `FALSE`.

`buffersize` The size in bytes for curl's receive buffer. Default is 16000 bytes or 16kb, which means it will use the default value of libcurl. Maximum buffersize in libcurl documentation is 512kb (or 512000 bytes), but curl package accepts 2147483647 bytes without returning errors.

`fresh_connect` If `TRUE`, opens a new curl connection for each IMAP command to be executed. Default is `NULL`. Nonetheless, most of mRpostman functions will set `fresh_connect = TRUE` for the retries in case of an connection error.

`timeout_ms` Time in miliseconds (ms) to wait until a connection or a command to be executed. Default is 5000ms (or 5 seconds). If a first execution is frustated, an error handler in each function (depending on `retires` value), will try to reconnect or re-execute the command.

`...` Further parameters added by mRpostman functions or IMAP parameters listed in `curl::curl_options`. Only for advanced users.

Value

An object of class `imapconf` containing the settings needed to connect to the IMAP server.

References

https://curl.haxx.se/libcurl/c/CURLOPT_BUFFERSIZE.html

Examples

```
## Not run:

# Gmail config example:
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

## End(Not run)
```

copy_msg *Copy Messages*

Description

Copies messages form a mailbox to another.

Usage

```
copy_msg(imapconf, msg_id, by = "MSN", to_mbox, reselect_mbox = FALSE,
  retries = 2)
```

Arguments

imapconf — An object of class imapconf generated with configure_imap and with a mbox item added after a mailbox selection with select_mailbox.

msg_id — A numeric vetor containing one or more messages ids.

by — Which id (MSN or UID) to use when searching for messages. "MSN" or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. "UID" or unique identifier is always the same during the life cycle of a message. Default is "MSN".

to_mbox — A character string specifying the mailbox to which messages will be copied to.

reselect_mbox — If TRUE, calls select_mailbox(mbox = to_mbox) before returning the output. Default is FALSE for moving and copying operations, whereas it is TRUE for renaming mailboxes.

retries — Number of attempts to connect and execute the command. Default is 2.

Value

An (invisible) list of length 2 containing the imapconf object and the previously inputed message ids (parameter msg_id).

See Also

Other miscellaneous: add_flags, count_msgs, delete_msg, expunge, get_max_id, get_min_id, move_msg, remove_flags, replace_flags

Examples

```

## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# copy search results from "Sent" to "INBOX"
results <- imapconf %>%
    select_mailbox(mbox = "Sent") %>%
   search_before(date_char = "10-may-2012") %$% #exposition pipe operator - pass two argg
    copy_msg(imapconf = imapconf, msg_id = msg_id, to_mbox = "INBOX")
```

```
## End(Not run)
```

count_msgs *Count the Number of Messages*

Description

Counts the number of messages with a respective flag in a mailbox.

Usage

```
count_msgs(imapconf, by = "MSN", flag, retries = 2)
```

Arguments

imapconf An object of class imapconf generated with configure_imap and with a mbox item added after a mailbox selection with select_mailbox.

by Which id (MSN or UID) to use when searching for messages. "MSN" or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. "UID" or unique identifier is always the same during the life cycle of a message. Default is "MSN".

flag A string specifying the flag to be used for filtering messages. Use flag_options to list the common flags used by IMAP servers.

retries Number of attempts to connect and execute the command. Default is 2.

Value

A numeric vector of length 1 containing the number of messages that meet the specified criteria.

See Also

Other miscellaneous: add_flags, copy_msg, delete_msg, expunge, get_max_id, get_min_id, move_msg, remove_flags, replace_flags

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
```

```
                              )

    results <- imapconf %>%
        select_mailbox(mbox = "INBOX") %>%
        count_msgs(by = "UID", flag = "SEEN")


    ## End(Not run)
```

`custom_search` *Custom Search*

Description

Allows a combination of several arguments using helper functions that serve as relational operators, such as OR and AND; and criteria helper functions such as before, since, on, sent_before, sent_since, sent_on, flag, string, smaller_than, larger_than, younger_than, or younger_than, in order to execute a custom search (with multiple arguments).

Usage

```
custom_search(imapconf, custom_request, negate = FALSE, by = "MSN",
  esearch = FALSE, return_imapconf = TRUE, retries = 2)
```

Arguments

`imapconf` An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.

`custom_request` A string directly specifying what to search or constructed by a combination of operators helper functions OR and AND, and criteria helper functions such as before, since, on, sent_before, sent_since, sent_on, flag, string, smaller_than, larger_than, younger_than, or younger_than.

`negate` If `TRUE`, negates the search and seeks for "NOT search_criterion". Default is `FALSE`.

`by` Which id (`MSN` or `UID`) to use when searching for messages. `"MSN"` or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. `"UID"` or unique identifier is always the same during the life cycle of a message. Default is `"MSN"`.

`esearch` A logical. Default is `FALSE`. If the IMAP server has `ESEARCH` capability, it can be used to optimize search results. It allows to condense results to message sets to cut down on transmission costs, e.g. `1:5` instead of writing ids individually such as `1,2,3,4,5`. It can be used along with buffersize to avoid results stripping. Check if your IMAP server supports `SEARCH` with `list_server_capabilities`.

`return_imapconf`
: A logical. If TRUE, the function returns a list of length 2, containing the imapconf object (IMAP settings) and the search results as message ids ("MSN" or "UID"). If FALSE, returns only the message ids as a numeric vector. Default is TRUE.

`retries`
: Number of attempts to connect and execute the command. Default is 2.

Value

Depending on the return_imapconf parameter, it can be a list of length 2 containing the imapconf object and message ids resulted from search, or a numeric vector containing only message ids.

Note

IMAP queries follows Polish notation, i.e. operators such as OR come before arguments, e.g. "OR argument1 argument2". Therefore, the relational operators functions in this package should be used like the following examples: `OR(before("17-Apr-2015"),string("FROM","Jim"))`. Even though there is no "AND" operator in IMAP, this package adds a helper funcion AND to indicate multiples arguments that must be searched together, e.g. `AND(since("01-Jul-2018"),smaller_than(16000))`.

See Also

Other Date-search operations: search_before, search_on, search_period, search_sent_before, search_sent_on, search_sent_period, search_sent_since, search_since

Other Size-search operations: search_larger_than, search_smaller_than

Other Flag-search operations: search_flag

Other Within-search operations: search_older_than, search_younger_than

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )
# search1
result1 <- imapconf %>%
    select_mailbox(mbox = "TAM") %>%
    custom_search(custom_request = AND(string(section_or_field = "TO",
                                              string = "allan"),
                                       before(date_char = "12-Apr-2009")
                                      ))
# searches for messages containing the string "allan" in the "TO" field
# and that were received before "12-Apr-2009".

# search2
```

```
result2 <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    custom_search(custom_request = OR(string(section_or_field = "TEXT",
                                             string = "Contract 2872827"),
                                      string(section_or_field = "Subject",
                                             string = "Buying operation")
                                             ))
# searches for messages containing the string "Contract 2872827" in the
# "TEXT" section or the string "Buying operation" in the "Subject" field
# of the HEADER.


## End(Not run)
```

delete_msg *Delete Messages*

Description

Deletes messages from a mailbox.

Usage

```
delete_msg(imapconf, msg_id, by = "MSN", retries = 2)
```

Arguments

`imapconf` An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.

`msg_id` A numeric vetor containing one or more messages ids.

`by` Which id (`MSN` or `UID`) to use when searching for messages. `"MSN"` or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. `"UID"` or unique identifier is always the same during the life cycle of a message. Default is `"MSN"`.

`retries` Number of attempts to connect and execute the command. Default is `2`.

Value

An (invisible) `list` of length `2` containing the `imapconf` object and the previously inputed message ids (parameter `msg_id`).

See Also

Other miscellaneous: `add_flags`, `copy_msg`, `count_msgs`, `expunge`, `get_max_id`, `get_min_id`, `move_msg`, `remove_flags`, `replace_flags`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# deleting based on search results
result1 <- imapconf %>%
    select_mailbox(mbox = "TAM") %>%
   search_before(date_char = "10-may-2012", by = "UID") %$% #modified pipe operator - pass two argg
    delete_msg(imapconf = imapconf, msg_id = msg_id)


# deleting a specific msg_id without a previous search
result2 <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    delete_msg(msg_id = 66128)


## End(Not run)
```

---

examine_mailbox *Examine Mailbox*

Description

Retrieve the number of recent messages and the total number of messages in a mailbox.

Usage

```
examine_mailbox(imapconf, retries = 2)
```

Arguments

imapconf	An object of class imapconf generated with configure_imap and with a mbox item added after a mailbox selection with select_mailbox
retries	Number of attempts to connect and execute the command. Default is 2.

Value

A vector containing the ("EXISTS") and the ("RECENT") number of messages in the selected mailbox.

See Also

Other mailbox commands: `list_mailboxes`, `list_server_capabilities`, `rename_mailbox`, `select_mailbox`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# list mailboxes
results <- imapconf %>%
  list_mailboxes() %>%
  select_mailbox(mbox = "Sent") %>%
  examine_mailbox()


## End(Not run)
```

---

expunge *Expunge Messages*

Description

Expunge a specific message (using specific UID) or all messages marked as "DELETED" in a mailbox.

Usage

```
expunge(imapconf, specific_UID = NULL, retries = 2)
```

Arguments

`imapconf`	An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.
`specific_UID`	A numeric vector containing one or more messages UIDs, if specific messages should be expunged and not the whole mailbox.
`retries`	Number of attempts to connect and execute the command. Default is `2`.

Value

The (invisible) imapconf object that was previously inputed.

See Also

Other miscellaneous: add_flags, copy_msg, count_msgs, delete_msg, get_max_id, get_min_id, move_msg, remove_flags, replace_flags

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# deleting a message using specific UID
result1 <- imapconf %>%
    select_mailbox(mbox = "TAM") %>%
    delete_msg(msg_id = 71171, by = "UID") %$%
    expunge(imapconf = imapconf, specific_UID = msg_id)


# expunge all message smarked as "DELETED" in INBOX
result2 <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    expunge()


## End(Not run)
```

fetch_full_msg *Fetch Full Messages*

Description

Fetch all the sections and fields of one or more messages.

Usage

```
fetch_full_msg(imapconf, msg_id, by = "MSN", peek = TRUE,
  partial = NULL, write_to_disk = FALSE, keep_in_mem = TRUE,
  retries = 2)
```

Arguments

`imapconf` An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.

`msg_id` A numeric vetor containing one or more messages ids.

`by` Which id (`MSN` or `UID`) to use when searching for messages. `"MSN"` or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. `"UID"` or unique identifier is always the same during the life cycle of a message. Default is `"MSN"`.

`peek` If `TRUE`, it does not mark messages as "read" after fetching. Default is `TRUE`.

`partial` `NULL` or a character string with format "startchar.endchar" indicating the size (in characters) of a message slice to fetch. Default is `NULL`, which fetchs the full specified content.

`write_to_disk` If `TRUE` writes fetch content of each message to the disk as a text file in the working directory. Default is `FALSE`.

`keep_in_mem` If `TRUE` keeps a copy of fetch results as an list in the R session when `write_to_disk` = `TRUE`. Default is `TRUE`. It can only be set `FALSE` when `write_to_disk` = `TRUE`.

`retries` Number of attempts to connect and execute the command. Default is `2`.

Value

A `list` or/and text files containing the fetch results.

See Also

Other fetch: `fetch_msg_header`, `fetch_msg_metadata`, `fetch_msg_text`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# fetching
results <- imapconf %>%
    select_mailbox(mbox = "TAM") %>%
    search_before(date_char = "10-may-2012", by = "UID") %$% #exposition pipe - two argg
    fetch_full_msg(imapconf = imapconf,
                   msg_id = msg_id,
                   by="UID", write_to_disk = TRUE,
                   keep_in_mem = FALSE,
                     partial = "0.789")


## End(Not run)
```

fetch_msg_header *Fetch Message Headers*

Description

Fetch header fields of one or more messages.

Usage

```
fetch_msg_header(imapconf, msg_id, by = "MSN", fields = NULL,
  negate_fields = FALSE, peek = TRUE, partial = NULL,
  write_to_disk = FALSE, keep_in_mem = TRUE, retries = 2)
```

Arguments

imapconf An object of class imapconf generated with configure_imap and with a mbox item added after a mailbox selection with select_mailbox.

msg_id A numeric vetor containing one or more messages ids.

by Which id (MSN or UID) to use when searching for messages. "MSN" or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. "UID" or unique identifier is always the same during the life cycle of a message. Default is "MSN".

fields A character vector specifying which field(s) to fetch from messages. See section_or_field_options

negate_fields If TRUE, negates the search and seeks for "NOT in the field". Default is FALSE.

peek If TRUE, it does not mark messages as "read" after fetching. Default is TRUE.

partial NULL or a character string with format "startchar.endchar" indicating the size (in characters) of a message slice to fetch. Default is NULL, which fetchs the full specified content.

write_to_disk If TRUE writes fetch content of each message to the disk as a text file in the working directory. Default is FALSE.

keep_in_mem If TRUE keeps a copy of fetch results as an list in the R session when write_to_disk = TRUE. Default is FALSE. It can only be set TRUE when write_to_disk = TRUE.

retries Number of attempts to connect and execute the command. Default is 2.

Value

A list or/and text files containing the fetch results.

See Also

Other fetch: `fetch_full_msg`, `fetch_msg_metadata`, `fetch_msg_text`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )
# fetching
results <- imapconf %>%
    select_mailbox(mbox = "TAM") %>%
    search_before(date_char = "10-may-2012", by = "UID") %$% #exposition pipe - two argg
    fetch_msg_header(imapconf = imapconf, msg_id = msg_id, fields = c("From", "To"))


## End(Not run)
```

`fetch_msg_metadata` *Fetch Message Metadata*

Description

Fetch Metadata of one or more messages.

Usage

```
fetch_msg_metadata(imapconf, msg_id, by = "MSN", metadata,
  write_to_disk = FALSE, keep_in_mem = TRUE, retries = 2)
```

Arguments

`imapconf`	An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.
`msg_id`	A numeric vetor containing one or more messages ids.
`by`	Which id (`MSN` or `UID`) to use when searching for messages. `"MSN"` or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. `"UID"` or unique identifier is always the same during the life cycle of a message. Default is `"MSN"`.

`metadata` A character vector or string specifying one or more items of the metadata of a message to fetch. See metadata_options.

`write_to_disk` If `TRUE` writes fetch content of each message to the disk as a text file in the working directory. Default is `FALSE`.

`keep_in_mem` If `TRUE` keeps a copy of fetch results as an list in the R session when `write_to_disk = TRUE`. Default is `FALSE`. It can only be set `TRUE` when `write_to_disk = TRUE`.

`retries` Number of attempts to connect and execute the command. Default is `2`.

Value

A `list` or/and text files containing the fetch results.

See Also

Other fetch: `fetch_full_msg`, `fetch_msg_header`, `fetch_msg_text`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# fetching
results <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    search_sent_since(date_char = "26-Aug-2008") %$% #exposition pipe - two argg
    fetch_msg_metadata(imapconf, msg_id = msg_id, by = "UID",
                     metadata = c("INTERNALDATE", "UID", "ENVELOPE", "FLAGS",
                                  "RFC822.SIZE", "BODYSTRUCTURE"))


## End(Not run)
```

`fetch_msg_text` *Fetch Message Text*

Description

Fetch Text section of one or more messages.

Usage

```
fetch_msg_text(imapconf, msg_id, by = "MSN", peek = TRUE,
  partial = NULL, write_to_disk = FALSE, keep_in_mem = TRUE,
  try_b64decode = FALSE, retries = 2)
```

Arguments

`imapconf` An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.

`msg_id` A numeric vetor containing one or more messages ids.

`by` Which id (`MSN` or `UID`) to use when searching for messages. `"MSN"` or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. `"UID"` or unique identifier is always the same during the life cycle of a message. Default is `"MSN"`.

`peek` If `TRUE`, it does not mark messages as "read" after fetching. Default is `TRUE`.

`partial` `NULL` or a character string with format "startchar.endchar" indicating the size (in characters) of a message slice to fetch. Default is `NULL`, which fetchs the full specified content.

`write_to_disk` If `TRUE` writes fetch content of each message to the disk as a text file in the working directory. Default is `FALSE`.

`keep_in_mem` If `TRUE` keeps a copy of fetch results as an list in the R session when `write_to_disk` = `TRUE`. Default is `FALSE`. It can only be set `TRUE` when `write_to_disk` = `TRUE`.

`try_b64decode` If `TRUE`, tries to guess and decode the fetched text from base64 format to `character`. Default is `FALSE`.

`retries` Number of attempts to connect and execute the command. Default is `2`.

Value

A `list` or/and text files containing the fetch results.

See Also

Other fetch: `fetch_full_msg`, `fetch_msg_header`, `fetch_msg_metadata`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )
```

```
# fetching and saving results to disk
results <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    search_sent_since(date_char = "26-Aug-2008") %$% #exposition pipe - two argg
    fetch_msg_text(imapconf = imapconf, msg_id = msg_id, write_to_disk = TRUE)


## End(Not run)
```

flag — *Criteria Helper Functions for Custom Search*

Description

Criteria helper functions to be combined inside a operator helper function as a custom request in custom_search.

Usage

```
flag(flag, negate = FALSE)
```

Arguments

flag	A string specifying the flag to be used for filtering messages. Use flag_options to list the flags available in your IMAP server.
negate	If TRUE, negates the search and seeks for "NOT search_criterion". Default is FALSE.

Value

A search string to be used as a custom_request parameter in custom_search function.

See Also

Other customsearch helper functions: AND, OR, before, larger_than, older_than, on, sent_before, sent_on, sent_since, since, smaller_than, string, younger_than

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
```

```
                              )

    # search
    result <- imapconf %>%
        select_mailbox(mbox = "INBOX") %>%
        custom_search(custom_request = AND(flag("UNSEEN"),
                                           smaller_than(size = 512000),
                                           negate = TRUE))
    # searches for messages with Flag "UNSEEN" AND NOT Smaller Than  512KB.


    ## End(Not run)
```

flag_options *Flag Options*

Description

Lists flags used by most of IMAP servers.

Usage

```
flag_options()
```

Value

A `data.frame` containing flag names, its descriptions, and its antonym versions.

Note

RFC 2060 lists a set of systems flags, which are presentend and extended by this function. We also present a list of negative flags. When using `verbose = TRUE` in configure_imap function, flags appear with "\", e.g. "\Seen". It is important to note that some (but not all) functions that deal with flags also accepts, keywords - words usually listed (when verbose = TRUE) together with flags by the server but with no leading "\"). This is the case of add_flags, remove_flags, and replace_flags.

References

Crispin, M., "Internet Message Access Protocol - Version 4rev1", RFC 2060, doi: 10.17487/RFC2060, December 1996, `https://www.rfc-editor.org/info/rfc2060`.

See Also

Other options: `metadata_options`, `section_or_field_options`

Examples

```
## Not run:

library(mRpostman)
flag_options()


## End(Not run)
```

get_attachments *Get Attachments*

Description

Get attachments after fetching full messages.

Usage

```
get_attachments(msg_list, content_disposition = "both")
```

Arguments

`msg_list` A `list` containing the full messages fetched from the server after using `fetch_full_msg`.

`content_disposition` A `string` indicating which type of "Content-Disposition" attachments should be retireved. Default is `"both"`, which retireves regular attachments ("Content-Disposition: attachment") and inline attachments ("Content-Disposition: inline").

Value

A logical `TRUE` if the attachments extraction was successful.

Note

All attachments will be stored in a folder labelled with the message id inside the working directory. This function currently handles only attachments encoded as `base64` text. It tries to guess all file extensions while decoding the text, but it may not be possible to do it in some circumstances. If it happens, you can try to change the file extension directly by renaming the file.

The "Content-Disposition" header specifies if the multipart electronic messages will be presented as a main document with a list of separate attachments ("Content-Disposition: attachment") or as a single document with the various parts displayed inline. The first requires positive action on the part of recipient, i.e. downloading the file, whereas inline components are displayed automatically when the message is viewed (*Troost, R., Dorner, S., and K. Moore, Ed. (1997)*). You can choose to download `both`, or only one of the types of attachments, using the argument `content_disposition`.

References

Troost, R., Dorner, S., and K. Moore, Ed., *Communicating Presentation Information in Internet Messages: The Content-Disposition Header Field*, RFC 2183, DOI 10.17487/RFC2183, August 1997, https://www.rfc-editor.org/info/rfc2183.

See Also

Other attachments: list_attachments

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# extracting attachments
imapconf %>%
    select_mailbox(mbox = "TAM") %>%
    search_before(date_char = "10-may-2012", by = "UID") %$% #exposition pipe - two argg
    fetch_full_msgs(imapconf = imapconf, msg_id = msg_id) %>%
    get_attachments()


## End(Not run)
```

get_max_id *Get Maximum Message ID in a Mailbox*

Description

Get the greater (maximum) message id in a Mailbox, considering messages with a specific flag.

Usage

```
get_max_id(imapconf, by = "MSN", flag, retries = 2)
```

Arguments

imapconf — An object of class imapconf generated with configure_imap and with a mbox item added after a mailbox selection with select_mailbox.

`by` Which id (`MSN` or `UID`) to use when searching for messages. `"MSN"` or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. `"UID"` or unique identifier is always the same during the life cycle of a message. Default is `"MSN"`.

`flag` A string specifying the flag to be used for filtering messages. Use flag_options to list the common flags used by IMAP servers. Default is `NULL`.

`retries` Number of attempts to connect and execute the command. Default is `2`.

Value

A numeric `vector` of length `1` indicating the minimum id.

Note

`get_max_id` uses `ESEARCH` extension. Check if your server supports `ESEARCH` capability with list_server_capabilities.

See Also

Other miscellaneous: `add_flags`, `copy_msg`, `count_msgs`, `delete_msg`, `expunge`, `get_min_id`, `move_msg`, `remove_flags`, `replace_flags`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

results <- imapconf %>%
    select_mailbox(mbox = "[Gmail]/Trash") %>%
    get_max_id(by = "UID", flag = "SEEN")


## End(Not run)
```

get_min_id *Get Minimum Message ID in a Mailbox*

Description

Get the smaller (minimum) message id in a Mailbox, considering messages with a specific flag.

Usage

```
get_min_id(imapconf, by = "MSN", flag, retries = 2)
```

Arguments

imapconf	An object of class imapconf generated with configure_imap and with a mbox item added after a mailbox selection with select_mailbox.
by	Which id (MSN or UID) to use when searching for messages. "MSN" or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. "UID" or unique identifier is always the same during the life cycle of a message. Default is "MSN".
flag	A string specifying the flag to be used for filtering messages. Use flag_options to list the common flags used by IMAP servers. Default is NULL.
retries	Number of attempts to connect and execute the command. Default is 2.

Value

A numeric vector of length 1 indicating the minimum id.

Note

get_max_id uses ESEARCH extension. Check if your server supports ESEARCH capability with list_server_capabilities.

See Also

Other miscellaneous: add_flags, copy_msg, count_msgs, delete_msg, expunge, get_max_id, move_msg, remove_flags, replace_flags

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                              username="your_username",
                              password=rstudioapi::askForPassword()
                             )

results <- imapconf %>%
    select_mailbox(mbox = "[Gmail]/Trash") %>%
    get_min_id(flag = "UNSEEN")


## End(Not run)
```

larger_than *Criteria Helper Functions for Custom Search*

Description

Criteria helper functions to be combined inside a operator helper function as a custom request in custom_search.

Usage

```
larger_than(size, negate = FALSE)
```

Arguments

size	An integer specifying the size (in number of characters) of message to be used as search criterion.
negate	If TRUE, negates the search and seeks for "NOT search_criterion". Default is FALSE.

Value

A search string to be used as a custom_request parameter in custom_search function.

See Also

Other customsearch helper functions: AND, OR, before, flag, older_than, on, sent_before, sent_on, sent_since, since, smaller_than, string, younger_than

Examples

```

## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                              password=rstudioapi::askForPassword()
                             )

# search
result <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    custom_search(custom_request =
                 OR(string(
                       section_or_field = "from", string = "allanvcq@gmail.com"),
                    larger_than(
                       size = 512000)
                   )
                 )
# searches for messages containing the string "allanvcq@gmail.com" in the
# field "FROM" OR those Larger Than 512KB.


## End(Not run)
```

list_attachments *List Attachments*

Description

List attachments filenames and Content-Disposition types after fetching full messages.

Usage

```
list_attachments(msg_list)
```

Arguments

msg_list	A `list` containing the full messages fetched from the server after using `fetch_full_msg`.

Value

A `list` of `data.frames` containing the filenames and its Content-Disposition types for each fetched message.

See Also

Other attachments: `get_attachments`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# listing attachments
attachments <- imapconf %>%
    select_mailbox(mbox = "TAM") %>%
    search_before(date_char = "10-may-2012", by = "UID") %$% #exposition pipe - two argg
    fetch_full_msg(imapconf = imapconf, msg_id = msg_id) %>%
    list_attachments()


## End(Not run)
```

`list_mailboxes` *Listing Mailboxes*

Description

List Mailboxes from IMAP server.

Usage

```
list_mailboxes(imapconf, retries = 2)
```

Arguments

`imapconf`	An object of class `imapconf` generated with configure_imap.
`retries`	Number of attempts to connect and execute the command. Default is `2`.

Value

A `list` containing the Mailboxes (root and children).

See Also

Other mailbox commands: `examine_mailbox`, `list_server_capabilities`, `rename_mailbox`, `select_mailbox`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                         )

# list mailboxes
results <- imapconf %>%
  list_mailboxes()


## End(Not run)
```

`list_server_capabilities`

IMAP Server Capabilities

Description

Lists IMAP server's capabilities.

Usage

```
list_server_capabilities(imapconf, retries = 2)
```

Arguments

`imapconf`	An object of class `imapconf` generated with configure_imap.
`retries`	Number of attempts to connect and execute the command. Default is 2.

Value

A character `vector` containing server's IMAP capacbilities.

See Also

Other mailbox commands: `examine_mailbox`, `list_mailboxes`, `rename_mailbox`, `select_mailbox`

Examples

```
## Not run:

# configure IMAP
#' library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                         )

# list server's capabilities
results <- imapconf %>%
  list_server_capabilities()


## End(Not run)
```

metadata_options *Message Metadata Options*

Description

List Metadata fields used in messages.

Usage

```
metadata_options()
```

Value

A `vector` containing message metadata fields.

Note

This function lists message metadata used by IMAP servers, according to the RFC 2060 (Crispin, 1996).

References

Crispin, M., "Internet Message Access Protocol - Version 4rev1", RFC 2060, doi: 10.17487/RFC2060, December 1996, `https://www.rfc-editor.org/info/rfc2060`.

See Also

Other options: `flag_options`, `section_or_field_options`

Examples

```
## Not run:

library(mRpostman)
metadata_options()


## End(Not run)
```

move_msg *Move Messages*

Description

Moves messages between mailboxes.

Usage

```
move_msg(imapconf, msg_id, by = "MSN", to_mbox, reselect_mbox = FALSE,
  retries = 2)
```

Arguments

imapconf	An object of class imapconf generated with configure_imap and with a mbox item added after a mailbox selection with select_mailbox.
msg_id	A numeric vetor containing one or more messages ids.
by	Which id (MSN or UID) to use when searching for messages. "MSN" or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. "UID" or unique identifier is always the same during the life cycle of a message. Default is "MSN".
to_mbox	A character string specifying the mailbox to which messages will be moved to.
reselect_mbox	If TRUE, calls select_mailbox(mbox = to_mbox) before returning the output. Default is FALSE for moving and copying operations, whereas it is TRUE for renaming mailboxes.
retries	Number of attempts to connect and execute the command. Default is 2.

Value

An (invisible) list of length 2 containing the imapconf object and the previously inputed message ids (parameter msg_id).

Note

move_msg uses MOVE extension. Check if your server supports MOVE capability with list_server_capabilities.

See Also

Other miscellaneous: add_flags, copy_msg, count_msgs, delete_msg, expunge, get_max_id, get_min_id, remove_flags, replace_flags

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# copy search results from "Sent" to "INBOX"
results <- imapconf %>%
    select_mailbox(mbox = "Sent") %>%
   search_before(date_char = "10-may-2012") %$% #exposition pipe operator - pass two argg
    move_msg(imapconf = imapconf, msg_id = msg_id, to_mbox = "Other Mailbox")

# the same result can be achieved with a combination of:
results <- imapconf %>%
    select_mailbox(mbox = "Sent") %>%
   search_before(date_char = "10-may-2012") %$% #exposition pipe operator - pass two argg
    copy_msg(imapconf = imapconf, msg_id = msg_id, to_mbox = "Other Mailbox") %$%
    add_flags(imapconf = imapconf, msg_id = msg_id, flags_to_set = "Deleted") %$%
    expunge()


## End(Not run)
```

older_than *Criteria Helper Functions for Custom Search*

Description

Criteria helper functions to be combined inside a operator helper function as a custom request in custom_search.

Usage

```
older_than(seconds, negate = FALSE)
```

Arguments

seconds — An integer specifying the number of seconds to be used as search criterion.

negate — If TRUE, negates the search and seeks for "NOT search_criterion". Default is FALSE.

Value

A search string to be used as a custom_request parameter in custom_search function.

See Also

Other customsearch helper functions: AND, OR, before, flag, larger_than, on, sent_before, sent_on, sent_since, since, smaller_than, string, younger_than

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search
result <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    custom_search(custom_request =
                 OR(string(
                       section_or_field = "from", string = "allanvcq@gmail.com"),
                    older_than(
                       seconds = 3600)
                   )
                 )
# searches for messages containing the string "allanvcq@gmail.com" in the
# field "FROM" OR messgaes that are "older" than one hour (messages arrived
# more than 3600 seconds ago).


## End(Not run)
```

on *Criteria Helper Functions for Custom Search*

Description

Criteria helper functions to be combined inside a operator helper function as a custom request in custom_search.

Usage

```
on(date_char, negate = FALSE)
```

Arguments

`date_char`	A character vector with format "DD-Mon-YYYY", e.g. "01-Apr-2019". We opted not to use objects of type "date", since IMAP servers like this not so common date format.
`negate`	If `TRUE`, negates the search and seeks for "NOT search_criterion". Default is `FALSE`.

Value

A search string to be used as a `custom_request` parameter in custom_search function.

See Also

Other customsearch helper functions: `AND`, `OR`, `before`, `flag`, `larger_than`, `older_than`, `sent_before`, `sent_on`, `sent_since`, `since`, `smaller_than`, `string`, `younger_than`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search
result <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    custom_search(custom_request = OR(on(date_char = "17-Apr-2019"),
                                  on(date_char = "17-Jul-2018")))
# searches for messages On "17-Apr-2019" OR On "17-Jul-2018".


## End(Not run)
```

OR *Relational Operators - Helper Functions for Custom Search*

Description

Allows a combination of criteria helper functions such as such as before, since, on, sent_before, sent_since, sent_on, flag, string, smaller_than, larger_than, younger_than, or younger_than, in order to execute a multiple criteria custom search.

Usage

```
OR(..., negate = FALSE)
```

Arguments

`...`	a combination of criteria helper functions with its arguments.
`negate`	If `TRUE`, negates the search and seeks for "NOT search_criterion". Default is `FALSE`.

Value

A search string to be used as a `custom_request` parameter in custom_search function.

See Also

Other customsearch helper functions: `AND`, `before`, `flag`, `larger_than`, `older_than`, `on`, `sent_before`, `sent_on`, `sent_since`, `since`, `smaller_than`, `string`, `younger_than`

Examples

```

## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search
result <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    custom_search(custom_request = OR(sent_since(date.char="17-Apr-2019"),
                                      smaller_than(size = 512000, negate = TRUE)
                                     ))
# searches for messages SentSince "17-Apr-2019" OR NOT Smaller Than  512KB.
```

```
## End(Not run)
```

remove_flags *Remove Flag(s) From Messages*

Description

Remove one or more flags from one or more messages.

Usage

```
remove_flags(imapconf, msg_id, by = "MSN", flags_to_unset, retries = 2)
```

Arguments

`imapconf` An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.

`msg_id` A numeric vetor containing one or more messages ids.

`by` Which id (`MSN` or `UID`) to use when searching for messages. `"MSN"` or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. `"UID"` or unique identifier is always the same during the life cycle of a message. Default is `"MSN"`.

`flags_to_unset` A character vector containing one ore more flag names to remove from the specified message ids.

`retries` Number of attempts to connect and execute the command. Default is `2`.

Value

An (invisible) `list` of length `2` containing the `imapconf` object and the previously inputed message ids (parameter `msg_id`).

Note

add_flags, remove_flags and replace_flags, accepts not only flags but also keywords (such as `$Phishing` on Gmail's IMAP server) to add, remove or replace.

IMAP servers do not allow setting the negative version of a flag when a message already has the positive version of it. If a message with `MSN 1` already has the "SEEN" flag, it is not allowed to add "UNSEEN" to that. Instead, you have to first remove the "SEEN" flag `remove_flags(imapconf,msg_id = 1,"SEEN")` and only then do `add_flags(imapconf,msg_id = 1,"SEEN")`. Another option is to complete override all the flags of a message or a set of messages using `replace_flags(imapconf,msg_id = 1,"SEEN")`.

See Also

Other miscellaneous: add_flags, copy_msg, count_msgs, delete_msg, expunge, get_max_id, get_min_id, move_msg, replace_flags

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

result1 <- imapconf %>%
    select_mailbox(mbox = "TAM") %>%
    search_before(date_char = "10-may-2012", by = "UID") %$% #exposition pipe - two argg
   remove_flags(imapconf = imapconf, msg_id = msg_id, flags_to_unset = c("RECENT", "$NotPhishing"))


## End(Not run)
```

---

rename_mailbox *Rename Mailbox*

Description

Rename a mailbox.

Usage

```
rename_mailbox(imapconf, new_name, reselect_mbox = TRUE, retries = 2)
```

Arguments

imapconf	An object of class imapconf generated with configure_imap and with a mbox item added after a mailbox selection with select_mailbox.
new_name	A string containing the new name to be set.
reselect_mbox	If TRUE, calls select_mailbox(mbox = to_mbox) before returning the output. Default is FALSE for moving and copying operations, whereas it is TRUE for renaming mailboxes.
retries	Number of attempts to connect and execute the command. Default is 2.

Value

An (invisible) imaconf object with the newer selected mailbox added to it. If reselect_mbox is FALSE, it returns the imaconf object with the former mailbox name.

See Also

Other mailbox commands: examine_mailbox, list_mailboxes, list_server_capabilities, select_mailbox

Examples

```
## Not run:

# configure IMAP
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# rename mailbox "Sent"
imapconf %>%
  select_mailbox(mbox = "Sent") %>%
  rename_mailbox(new_name = "Sent Mail")


## End(Not run)
```

---

replace_flags *Replace Flag(s) in Messages*

Description

Replaces all flags in messages by one or more flags.

Usage

```
replace_flags(imapconf, msg_id, by = "MSN", flags_to_set, retries = 2)
```

Arguments

imapconf — An object of class imapconf generated with configure_imap and with a mbox item added after a mailbox selection with select_mailbox.

msg_id — A numeric vetor containing one or more messages ids.

by — Which id (MSN or UID) to use when searching for messages. "MSN" or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. "UID" or unique identifier is always the same during the life cycle of a message. Default is "MSN".

`flags_to_set` A character vector containing one ore more flag names to add to or replace in the specified message ids.

`retries` Number of attempts to connect and execute the command. Default is `2`.

Value

An (invisible) `list` of length `2` containing the `imapconf` object and the previously inputed message ids (parameter `msg_id`).

Note

add_flags, remove_flags and replace_flags, accepts not only flags but also keywords (such as `$Phishing` on Gmail's IMAP server) to add, remove or replace.

IMAP servers do not allow setting the negative version of a flag when a message already has the positive version of it. If a message with `MSN 1` already has the "SEEN" flag, it is not allowed to add "UNSEEN" to that. Instead, you have to first remove the "SEEN" flag `remove_flags(imapconf,msg_id = 1,"SEEN")` and only then do `add_flags(imapconf,msg_id = 1,"SEEN")`. Another option is to complete override all the flags of a message or a set of messages using `replace_flags(imapconf,msg_id = 1,"SEEN")`.

See Also

Other miscellaneous: `add_flags`, `copy_msg`, `count_msgs`, `delete_msg`, `expunge`, `get_max_id`, `get_min_id`, `move_msg`, `remove_flags`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

result1 <- imapconf %>%
    select_mailbox(mbox = "TAM") %>%
    search_before(date_char = "10-may-2012", by = "UID") %$% #exposition pipe - two argg
   replace_flags(imapconf = imapconf, msg_id = msg_id, flags_to_set = c("SEEN", "DRAFT"))


## End(Not run)
```

search_before *Search By Internal Date*

Description

Functions that allows searching for messages using the (internal) date criteria, such as before, since, on, and period.

Usage

```
search_before(imapconf, date_char, negate = FALSE, by = "MSN",
  flag = NULL, esearch = FALSE, return_imapconf = TRUE,
  retries = 2)
```

Arguments

`imapconf` An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.

`date_char` A character vector with format "DD-Mon-YYYY", e.g. "01-Apr-2019". We opted not to use objects of type "date", since IMAP servers like this not so common date format.

`negate` If `TRUE`, negates the search and seeks for "NOT search_criterion". Default is `FALSE`.

`by` Which id (`MSN` or `UID`) to use when searching for messages. `"MSN"` or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. `"UID"` or unique identifier is always the same during the life cycle of a message. Default is `"MSN"`.

`flag` Optional parameter that adds a flag filter to the search. Use flag_options to list the common flags used by IMAP servers. Default is `NULL`.

`esearch` A logical. Default is `FALSE`. If the IMAP server has `ESEARCH` capability, it can be used to optimize search results. It allows to condense results to message sets to cut down on transmission costs, e.g. `1:5` instead of writing ids individually such as `1,2,3,4,5`. It can be used along with buffersize to avoid results stripping. Check if your IMAP server supports `SEARCH` with `list_server_capabilities`.

`return_imapconf`
A logical. If `TRUE`, the function returns a `list` of length `2`, containing the `imapconf` object (IMAP settings) and the search results as message ids (`"MSN"` or `"UID"`). If `FALSE`, returns only the message ids as a numeric vector. Default is `TRUE`.

`retries` Number of attempts to connect and execute the command. Default is `2`.

Value

Depending on the `return_imapconf` parameter, it can be a `list` of length 2 containing the `imapconf` object and message ids resulted from search, or a numeric vector containing only message ids.

Note

search_before, search_since, search_on, and search_period use internal date, which reflects the moment when the message was received. search_sent_before, search_sent_since, search_sent_on, and search_sent_period use RFC-2822 date header (origination date), which "specifies the date and time at which the creator of the message indicated that the message was complete and ready to enter the mail delivery system" (Resnick, 2008). Dates in both methods must be the same most of time. Nonetheless, using internal date for search is faster (Babcock, 2016).

References

Resnick, P., Ed., "Internet Message Format", RFC 5322, doi: 10.17487/RFC5322, October 2008, `https://www.rfc-editor.org/info/rfc5322`.

Babcock, N., "Introduction to IMAP", Blog, May 2016, `https://nbsoftsolutions.com/blog/introduction-to-imap`.

See Also

Other Date-search operations: `custom_search`, `search_on`, `search_period`, `search_sent_before`, `search_sent_on`, `search_sent_period`, `search_sent_since`, `search_since`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search 1
result1 <- imapconf %>%
    select_mailbox(mbox = "[Gmail]/Sent") %>%
    search_before(date_char = "17-Apr-2012")

# search 2 (brazilian portuguese example -- mailboxes with compound names)
result2 <- imapconf %>%
    select_mailbox(mbox = "[Gmail]/Com estrela") %>%
    search_before(date_char = "17-Dec-2018", flag = "ANSWERED")
# depending on results, it might be necessary to increase buffersize in
# configure_imap()
```

```
## End(Not run)
```

search_flag *Search By Flag*

Description

Searches for messages marked with a specific flag.

Usage

```
search_flag(imapconf, flag, negate = FALSE, by = "MSN",
  esearch = FALSE, return_imapconf = TRUE, retries = 2)
```

Arguments

`imapconf` An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.

`flag` A string specifying the flag to be used for filtering messages. Use flag_options to list the common flags used by IMAP servers.

`negate` If `TRUE`, negates the search and seeks for "NOT search_criterion". Default is `FALSE`.

`by` Which id (`MSN` or `UID`) to use when searching for messages. `"MSN"` or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. `"UID"` or unique identifier is always the same during the life cycle of a message. Default is `"MSN"`.

`esearch` A logical. Default is `FALSE`. If the IMAP server has `ESEARCH` capability, it can be used to optimize search results. It allows to condense results to message sets to cut down on transmission costs, e.g. `1:5` instead of writing ids individually such as `1,2,3,4,5`. It can be used along with buffersize to avoid results stripping. Check if your IMAP server supports `SEARCH` with `list_server_capabilities`.

`return_imapconf` A logical. If `TRUE`, the function returns a `list` of length `2`, containing the `imapconf` object (IMAP settings) and the search results as message ids (`"MSN"` or `"UID"`). If `FALSE`, returns only the message ids as a numeric vector. Default is `TRUE`.

`retries` Number of attempts to connect and execute the command. Default is `2`.

Value

Depending on the `return_imapconf` parameter, it can be a `list` of length `2` containing the `imapconf` object and message ids resulted from search, or a numeric vector containing only message ids.

See Also

Other Flag-search operations: `custom_search`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search
results <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    search_flag(flag = "RECENT", by = "UID")


## End(Not run)
```

search_larger_than *Search By Syze*

Description

Searches for messages by bytes size.

Usage

```
search_larger_than(imapconf, size, negate = FALSE, by = "MSN",
  flag = NULL, esearch = FALSE, return_imapconf = TRUE,
  retries = 2)
```

Arguments

`imapconf` An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.

`size` An integer specifying the size (in number of characters) of message to be used as search criterion.

`negate` If `TRUE`, negates the search and seeks for "NOT search_criterion". Default is `FALSE`.

by
: Which id (MSN or UID) to use when searching for messages. "MSN" or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. "UID" or unique identifier is always the same during the life cycle of a message. Default is "MSN".

flag
: Optional parameter that adds a flag filter to the search. Use flag_options to list the common flags used by IMAP servers. Default is NULL.

esearch
: A logical. Default is FALSE. If the IMAP server has ESEARCH capability, it can be used to optimize search results. It allows to condense results to message sets to cut down on transmission costs, e.g. 1:5 instead of writing ids individually such as 1,2,3,4,5. It can be used along with buffersize to avoid results stripping. Check if your IMAP server supports SEARCH with list_server_capabilities.

return_imapconf
: A logical. If TRUE, the function returns a list of length 2, containing the imapconf object (IMAP settings) and the search results as message ids ("MSN" or "UID"). If FALSE, returns only the message ids as a numeric vector. Default is TRUE.

retries
: Number of attempts to connect and execute the command. Default is 2.

Value

Depending on the return_imapconf parameter, it can be a list of length 2 containing the imapconf object and message ids resulted from search, or a numeric vector containing only message ids.

See Also

Other Size-search operations: custom_search, search_smaller_than

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search
results <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    search_larger_than(size = 512000, flag = "UNSEEN") # larger than 512KB


## End(Not run)
```

search_older_than *Search Within seconds*

Description

Searches for messages within a number of seconds from the server's current time to fetch messages.

Usage

```
search_older_than(imapconf, seconds, negate = FALSE, by = "MSN",
  flag = NULL, esearch = FALSE, return_imapconf = TRUE,
  retries = 2)
```

Arguments

`imapconf` An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.

`seconds` An integer specifying the number of seconds to be used as search criterion.

`negate` If `TRUE`, negates the search and seeks for "NOT search_criterion". Default is `FALSE`.

`by` Which id (`MSN` or `UID`) to use when searching for messages. `"MSN"` or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. `"UID"` or unique identifier is always the same during the life cycle of a message. Default is `"MSN"`.

`flag` Optional parameter that adds a flag filter to the search. Use flag_options to list the common flags used by IMAP servers. Default is `NULL`.

`esearch` A logical. Default is `FALSE`. If the IMAP server has `ESEARCH` capability, it can be used to optimize search results. It allows to condense results to message sets to cut down on transmission costs, e.g. `1:5` instead of writing ids individually such as `1,2,3,4,5`. It can be used along with buffersize to avoid results stripping. Check if your IMAP server supports `SEARCH` with `list_server_capabilities`.

`return_imapconf`
A logical. If `TRUE`, the function returns a `list` of length `2`, containing the `imapconf` object (IMAP settings) and the search results as message ids (`"MSN"` or `"UID"`). If `FALSE`, returns only the message ids as a numeric vector. Default is `TRUE`.

`retries` Number of attempts to connect and execute the command. Default is `2`.

Value

Depending on the `return_imapconf` parameter, it can be a `list` of length `2` containing the `imapconf` object and message ids resulted from search, or a numeric vector containing only message ids.

See Also

Other Within-search operations: `custom_search`, `search_younger_than`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search
results <- imapconf %>%
    select_mailbox(mbox = "Smiles GOL") %>%
    search_older_than(seconds = 3600)


## End(Not run)
```

`search_on` *Search By Internal Date*

Description

Functions that allows searching for messages using the (internal) date criteria, such as before, since, on, and period.

Usage

```
search_on(imapconf, date_char, negate = FALSE, by = "MSN",
  flag = NULL, esearch = FALSE, return_imapconf = TRUE,
  retries = 2)
```

Arguments

`imapconf` An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.

`date_char` A character vector with format "DD-Mon-YYYY", e.g. "01-Apr-2019". We opted not to use objects of type "date", since IMAP servers like this not so common date format.

`negate` If `TRUE`, negates the search and seeks for "NOT search_criterion". Default is `FALSE`.

`by` Which id (`MSN` or `UID`) to use when searching for messages. `"MSN"` or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. `"UID"` or unique identifier is always the same during the life cycle of a message. Default is `"MSN"`.

`flag` Optional parameter that adds a flag filter to the search. Use flag_options to list the common flags used by IMAP servers. Default is `NULL`.

`esearch` A logical. Default is `FALSE`. If the IMAP server has `ESEARCH` capability, it can be used to optimize search results. It allows to condense results to message sets to cut down on transmission costs, e.g. `1:5` instead of writing ids individually such as `1,2,3,4,5`. It can be used along with buffersize to avoid results stripping. Check if your IMAP server supports `SEARCH` with `list_server_capabilities`.

`return_imapconf`

A logical. If `TRUE`, the function returns a `list` of length `2`, containing the `imapconf` object (IMAP settings) and the search results as message ids (`"MSN"` or `"UID"`). If `FALSE`, returns only the message ids as a numeric vector. Default is `TRUE`.

`retries` Number of attempts to connect and execute the command. Default is `2`.

Value

Depending on the `return_imapconf` parameter, it can be a `list` of length `2` containing the `imapconf` object and message ids resulted from search, or a numeric vector containing only message ids.

References

Resnick, P., Ed., "Internet Message Format", RFC 5322, doi: 10.17487/RFC5322, October 2008, `https://www.rfc-editor.org/info/rfc5322`.

Babcock, N., "Introduction to IMAP", Blog, May 2016, `https://nbsoftsolutions.com/blog/introduction-to-imap`.

See Also

Other Date-search operations: `custom_search`, `search_before`, `search_period`, `search_sent_before`, `search_sent_on`, `search_sent_period`, `search_sent_since`, `search_since`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )
```

```
# list mailboxes
results <- imapconf %>%
    list_mailboxes()

# search 1
result1 <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    search_on(date_char = "17-Dec-2003")

# search 2 (mailboxes with compound names)
result2 <- imapconf %>%
    select_mailbox(mbox = "Special Mailbox") %>%
    search_on(date_char = "17-Sep-2018", flag = "UNDELETED")
# depending on results, it might be necessary to increase buffersize in
# configure_imap()


## End(Not run)
```

`search_period` *Search By Internal Date*

Description

Functions that allows searching for messages using the (internal) date criteria, such as before, since, on, and period.

Usage

```
search_period(imapconf, since_date_char, before_date_char,
  negate = FALSE, by = "MSN", flag = NULL, esearch = FALSE,
  return_imapconf = TRUE, retries = 2)
```

Arguments

`imapconf` An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.

`since_date_char` A character vector with format "DD-Mon-YYYY", e.g. "01-Apr-2019" indicating the start date (inclusive). We opted not to use objects of type "date", since IMAP servers like this not so common date format.

`before_date_char` A character vector with format "DD-Mon-YYYY", e.g. "01-Apr-2019" indicating the stop date (exclusive). We opted not to use objects of type "date", since IMAP server likes this not so common date format.

`negate` If `TRUE`, negates the search and seeks for "NOT search_criterion". Default is `FALSE`.

`by` Which id (`MSN` or `UID`) to use when searching for messages. `"MSN"` or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. `"UID"` or unique identifier is always the same during the life cycle of a message. Default is `"MSN"`.

`flag` Optional parameter that adds a flag filter to the search. Use flag_options to list the common flags used by IMAP servers. Default is `NULL`.

`esearch` A logical. Default is `FALSE`. If the IMAP server has `ESEARCH` capability, it can be used to optimize search results. It allows to condense results to message sets to cut down on transmission costs, e.g. `1:5` instead of writing ids individually such as `1,2,3,4,5`. It can be used along with buffersize to avoid results stripping. Check if your IMAP server supports `SEARCH` with `list_server_capabilities`.

`return_imapconf`
A logical. If `TRUE`, the function returns a `list` of length `2`, containing the `imapconf` object (IMAP settings) and the search results as message ids (`"MSN"` or `"UID"`). If `FALSE`, returns only the message ids as a numeric vector. Default is `TRUE`.

`retries` Number of attempts to connect and execute the command. Default is `2`.

Value

Depending on the `return_imapconf` parameter, it can be a `list` of length `2` containing the `imapconf` object and message ids resulted from search, or a numeric vector containing only message ids.

Note

search_before, search_since, search_on, and search_period use internal date, which reflects the moment when the message was received. search_sent_before, search_sent_since, search_sent_on, and search_sent_period use RFC-2822 date header (origination date), which "specifies the date and time at which the creator of the message indicated that the message was complete and ready to enter the mail delivery system" (Resnick, 2008). Dates in both methods must be the same most of time. Nonetheless, using internal date for search is faster (Babcock, 2016).

References

Resnick, P., "Internet Message Format", RFC 5322, October 2008.

Babcock, N., "Introduction to IMAP", Blog, May 2016.

See Also

Other Date-search operations: `custom_search`, `search_before`, `search_on`, `search_sent_before`, `search_sent_on`, `search_sent_period`, `search_sent_since`, `search_since`

Examples

```
## Not run:
```

```
# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )


# search
results <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    search_period(since_date_char = "17-Apr-2012",
                  before_date_char = "30-Jun-2015",
                  flag = "DRAFT")



## End(Not run)
```

search_sent_before *Search By Origination (RC-2822 Header) Date*

Description

Functions that allows searching for messages using the (RC-2822 Header) date criteria, such as before, since, on, and period.

Usage

```
search_sent_before(imapconf, date_char, negate = FALSE, by = "MSN",
  flag = NULL, esearch = FALSE, return_imapconf = TRUE,
  retries = 2)
```

Arguments

`imapconf` An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.

`date_char` A character vector with format "DD-Mon-YYYY", e.g. "01-Apr-2019". We opted not to use objects of type "date", since IMAP servers like this not so common date format.

`negate` If `TRUE`, negates the search and seeks for "NOT search_criterion". Default is `FALSE`.

`by` Which id (`MSN` or `UID`) to use when searching for messages. `"MSN"` or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap.

"UID" or unique identifier is always the same during the life cycle of a message. Default is "MSN".

flag — Optional parameter that adds a flag filter to the search. Use flag_options to list the common flags used by IMAP servers. Default is NULL.

esearch — A logical. Default is FALSE. If the IMAP server has ESEARCH capability, it can be used to optimize search results. It allows to condense results to message sets to cut down on transmission costs, e.g. 1:5 instead of writing ids individually such as 1,2,3,4,5. It can be used along with buffersize to avoid results stripping. Check if your IMAP server supports SEARCH with list_server_capabilities.

return_imapconf — A logical. If TRUE, the function returns a list of length 2, containing the imapconf object (IMAP settings) and the search results as message ids ("MSN" or "UID"). If FALSE, returns only the message ids as a numeric vector. Default is TRUE.

retries — Number of attempts to connect and execute the command. Default is 2.

Value

Depending on the return_imapconf parameter, it can be a list of length 2 containing the imapconf object and message ids resulted from search, or a numeric vector containing only message ids.

Note

search_before, search_since, search_on, and search_period use internal date, which reflects the moment when the message was received. search_sent_before, search_sent_since, search_sent_on, and search_sent_period use RFC-2822 date header (origination date), which "specifies the date and time at which the creator of the message indicated that the message was complete and ready to enter the mail delivery system" (Resnick, 2008). Dates in both methods must be the same most of time. Nonetheless, using internal date for search is faster (Babcock, 2016).

References

Resnick, P., "Internet Message Format", RFC 5322, October 2008.

Babcock, N., "Introduction to IMAP", Blog, May 2016.

See Also

Other Date-search operations: custom_search, search_before, search_on, search_period, search_sent_on, search_sent_period, search_sent_since, search_since

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search
results <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    search_sent_before(date_char = "17-Apr-2012",
                      negate = TRUE,
                      flag = "UNANSWERED")


## End(Not run)
```

---

search_sent_on *Search By Origination (RC-2822 Header) Date*

Description

Functions that allows searching for messages using the (RC-2822 Header) date criteria, such as before, since, on, and period.

Usage

```
search_sent_on(imapconf, date_char, negate = FALSE, by = "MSN",
  flag = NULL, esearch = FALSE, return_imapconf = TRUE,
  retries = 2)
```

Arguments

`imapconf` An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.

`date_char` A character vector with format "DD-Mon-YYYY", e.g. "01-Apr-2019". We opted not to use objects of type "date", since IMAP servers like this not so common date format.

`negate` If `TRUE`, negates the search and seeks for "NOT search_criterion". Default is `FALSE`.

`by` Which id (`MSN` or `UID`) to use when searching for messages. `"MSN"` or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. `"UID"` or unique identifier is always the same during the life cycle of a message. Default is `"MSN"`.

`flag` Optional parameter that adds a flag filter to the search. Use flag_options to list the common flags used by IMAP servers. Default is `NULL`.

`esearch` A logical. Default is `FALSE`. If the IMAP server has `ESEARCH` capability, it can be used to optimize search results. It allows to condense results to message sets to cut down on transmission costs, e.g. `1:5` instead of writing ids individually such as `1,2,3,4,5`. It can be used along with buffersize to avoid results stripping. Check if your IMAP server supports `SEARCH` with `list_server_capabilities`.

`return_imapconf` A logical. If `TRUE`, the function returns a `list` of length `2`, containing the `imapconf` object (IMAP settings) and the search results as message ids (`"MSN"` or `"UID"`). If `FALSE`, returns only the message ids as a numeric vector. Default is `TRUE`.

`retries` Number of attempts to connect and execute the command. Default is `2`.

Value

Depending on the `return_imapconf` parameter, it can be a `list` of length `2` containing the `imapconf` object and message ids resulted from search, or a numeric vector containing only message ids.

References

Resnick, P., "Internet Message Format", RFC 5322, October 2008.

Babcock, N., "Introduction to IMAP", Blog, May 2016.

See Also

Other Date-search operations: `custom_search`, `search_before`, `search_on`, `search_period`, `search_sent_before`, `search_sent_period`, `search_sent_since`, `search_since`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search 1
result1 <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    search_sent_on(date_char = "17-Dec-2003")

# search 2 (mailboxes with compound names)
result2 <- imapconf %>%
    select_mailbox(mbox = "Special Mailbox") %>%
    search_sent_on(date_char = "17-Sep-2018", flag = "UNDELETED")
```

```
## End(Not run)
```

search_sent_period *Search By Origination (RC-2822 Header) Date*

Description

Functions that allows searching for messages using the (RC-2822 Header) date criteria, such as before, since, on, and period.

Usage

```
search_sent_period(imapconf, since_date_char, before_date_char,
  negate = FALSE, by = "MSN", flag = NULL, esearch = FALSE,
  return_imapconf = TRUE, retries = 2)
```

Arguments

`imapconf`	An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.
`since_date_char`	A character vector with format "DD-Mon-YYYY", e.g. "01-Apr-2019" indicating the start date (inclusive). We opted not to use objects of type "date", since IMAP servers like this not so common date format.
`before_date_char`	A character vector with format "DD-Mon-YYYY", e.g. "01-Apr-2019" indicating the stop date (exclusive). We opted not to use objects of type "date", since IMAP server likes this not so common date format.
`negate`	If `TRUE`, negates the search and seeks for "NOT search_criterion". Default is `FALSE`.
`by`	Which id (`MSN` or `UID`) to use when searching for messages. `"MSN"` or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. `"UID"` or unique identifier is always the same during the life cycle of a message. Default is `"MSN"`.
`flag`	Optional parameter that adds a flag filter to the search. Use flag_options to list the common flags used by IMAP servers. Default is `NULL`.
`esearch`	A logical. Default is `FALSE`. If the IMAP server has `ESEARCH` capability, it can be used to optimize search results. It allows to condense results to message sets to cut down on transmission costs, e.g. `1:5` instead of writing ids individually such as `1,2,3,4,5`. It can be used along with buffersize to avoid results stripping. Check if your IMAP server supports `SEARCH` with `list_server_capabilities`.

`return_imapconf`
A logical. If `TRUE`, the function returns a `list` of length `2`, containing the `imapconf` object (IMAP settings) and the search results as message ids (`"MSN"` or `"UID"`). If `FALSE`, returns only the message ids as a numeric vector. Default is `TRUE`.

`retries` Number of attempts to connect and execute the command. Default is `2`.

Value

Depending on the `return_imapconf` parameter, it can be a `list` of length `2` containing the `imapconf` object and message ids resulted from search, or a numeric vector containing only message ids.

Note

search_before, search_since, search_on, and search_period use internal date, which reflects the moment when the message was received. search_sent_before, search_sent_since, search_sent_on, and search_sent_period use RFC-2822 date header (origination date), which "specifies the date and time at which the creator of the message indicated that the message was complete and ready to enter the mail delivery system" (Resnick, 2008). Dates in both methods must be the same most of time. Nonetheless, using internal date for search is faster (Babcock, 2016).

References

Resnick, P., "Internet Message Format", RFC 5322, October 2008.

Babcock, N., "Introduction to IMAP", Blog, May 2016.

See Also

Other Date-search operations: `custom_search`, `search_before`, `search_on`, `search_period`, `search_sent_before`, `search_sent_on`, `search_sent_since`, `search_since`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search
results <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    search_sent_period(since_date_char = "17-Apr-2012",
                 before_date_char = "30-Jun-2015",
                 flag = "UNFLAGGED")
```

```
## End(Not run)
```

search_sent_since *Search By Origination (RC-2822 Header) Date*

Description

Functions that allows searching for messages using the (RC-2822 Header) date criteria, such as before, since, on, and period.

Usage

```
search_sent_since(imapconf, date_char, negate = FALSE, by = "MSN",
  flag = NULL, esearch = FALSE, return_imapconf = TRUE,
  retries = 2)
```

Arguments

`imapconf` An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.

`date_char` A character vector with format "DD-Mon-YYYY", e.g. "01-Apr-2019". We opted not to use objects of type "date", since IMAP servers like this not so common date format.

`negate` If `TRUE`, negates the search and seeks for "NOT search_criterion". Default is `FALSE`.

`by` Which id (`MSN` or `UID`) to use when searching for messages. `"MSN"` or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. `"UID"` or unique identifier is always the same during the life cycle of a message. Default is `"MSN"`.

`flag` Optional parameter that adds a flag filter to the search. Use flag_options to list the common flags used by IMAP servers. Default is `NULL`.

`esearch` A logical. Default is `FALSE`. If the IMAP server has `ESEARCH` capability, it can be used to optimize search results. It allows to condense results to message sets to cut down on transmission costs, e.g. `1:5` instead of writing ids individually such as `1,2,3,4,5`. It can be used along with buffersize to avoid results stripping. Check if your IMAP server supports `SEARCH` with `list_server_capabilities`.

`return_imapconf`
A logical. If `TRUE`, the function returns a `list` of length `2`, containing the `imapconf` object (IMAP settings) and the search results as message ids (`"MSN"` or `"UID"`). If `FALSE`, returns only the message ids as a numeric vector. Default is `TRUE`.

`retries` Number of attempts to connect and execute the command. Default is `2`.

Value

Depending on the `return_imapconf` parameter, it can be a `list` of length 2 containing the `imapconf` object and message ids resulted from search, or a numeric vector containing only message ids.

References

Resnick, P., "Internet Message Format", RFC 5322, October 2008.

Babcock, N., "Introduction to IMAP", Blog, May 2016.

See Also

Other Date-search operations: `custom_search`, `search_before`, `search_on`, `search_period`, `search_sent_before`, `search_sent_on`, `search_sent_period`, `search_since`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search 1
result1 <- imapconf %>%
    select_mailbox(mbox = "Sent") %>%
    search_sent_since(date_char = "17-Apr-2012")

# search 2
result2 <- imapconf %>%
    select_mailbox(mbox = "Inbox") %>%
    search_sent_since(date_char = "17-Jun-2019", flag = "FLAGGED")


## End(Not run)
```

`search_since` *Search By Internal Date*

Description

Functions that allows searching for messages using the (internal) date criteria, such as before, since, on, and period.

Usage

```
search_since(imapconf, date_char, negate = FALSE, by = "MSN",
  flag = NULL, esearch = FALSE, return_imapconf = TRUE,
  retries = 2)
```

Arguments

`imapconf` An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.

`date_char` A character vector with format "DD-Mon-YYYY", e.g. "01-Apr-2019". We opted not to use objects of type "date", since IMAP servers like this not so common date format.

`negate` If `TRUE`, negates the search and seeks for "NOT search_criterion". Default is `FALSE`.

`by` Which id (`MSN` or `UID`) to use when searching for messages. `"MSN"` or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. `"UID"` or unique identifier is always the same during the life cycle of a message. Default is `"MSN"`.

`flag` Optional parameter that adds a flag filter to the search. Use flag_options to list the common flags used by IMAP servers. Default is `NULL`.

`esearch` A logical. Default is `FALSE`. If the IMAP server has `ESEARCH` capability, it can be used to optimize search results. It allows to condense results to message sets to cut down on transmission costs, e.g. `1:5` instead of writing ids individually such as `1,2,3,4,5`. It can be used along with buffersize to avoid results stripping. Check if your IMAP server supports `SEARCH` with `list_server_capabilities`.

`return_imapconf`
A logical. If `TRUE`, the function returns a `list` of length `2`, containing the `imapconf` object (IMAP settings) and the search results as message ids (`"MSN"` or `"UID"`). If `FALSE`, returns only the message ids as a numeric vector. Default is `TRUE`.

`retries` Number of attempts to connect and execute the command. Default is `2`.

Value

Depending on the `return_imapconf` parameter, it can be a `list` of length 2 containing the `imapconf` object and message ids resulted from search, or a numeric vector containing only message ids.

References

Resnick, P., Ed., "Internet Message Format", RFC 5322, doi: 10.17487/RFC5322, October 2008, `https://www.rfc-editor.org/info/rfc5322`.

Babcock, N., "Introduction to IMAP", Blog, May 2016, `https://nbsoftsolutions.com/blog/introduction-to-imap`.

See Also

Other Date-search operations: custom_search, search_before, search_on, search_period, search_sent_before, search_sent_on, search_sent_period, search_sent_since

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search 1
result1 <- imapconf %>%
    select_mailbox(mbox = "Sent") %>%
    search_since(date_char = "17-Apr-2012")

# search 2
result2 <- imapconf %>%
    select_mailbox(mbox = "Inbox") %>%
    search_since(date_char = "17-Jun-2019", flag = "FLAGGED")
# depending on results, it might be necessary to increase buffersize in
# configure_imap()


## End(Not run)
```

search_smaller_than *Search By Syze*

Description

Searches for messages by bytes size.

Usage

```
search_smaller_than(imapconf, size, negate = FALSE, by = "MSN",
  flag = NULL, esearch = FALSE, return_imapconf = TRUE,
  retries = 2)
```

Arguments

`imapconf` An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.

`size` An integer specifying the size (in number of characters) of message to be used as search criterion.

`negate` If `TRUE`, negates the search and seeks for "NOT search_criterion". Default is `FALSE`.

`by` Which id (`MSN` or `UID`) to use when searching for messages. `"MSN"` or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. `"UID"` or unique identifier is always the same during the life cycle of a message. Default is `"MSN"`.

`flag` Optional parameter that adds a flag filter to the search. Use flag_options to list the common flags used by IMAP servers. Default is `NULL`.

`esearch` A logical. Default is `FALSE`. If the IMAP server has `ESEARCH` capability, it can be used to optimize search results. It allows to condense results to message sets to cut down on transmission costs, e.g. `1:5` instead of writing ids individually such as `1,2,3,4,5`. It can be used along with buffersize to avoid results stripping. Check if your IMAP server supports `SEARCH` with `list_server_capabilities`.

`return_imapconf` A logical. If `TRUE`, the function returns a `list` of length `2`, containing the `imapconf` object (IMAP settings) and the search results as message ids (`"MSN"` or `"UID"`). If `FALSE`, returns only the message ids as a numeric vector. Default is `TRUE`.

`retries` Number of attempts to connect and execute the command. Default is `2`.

Value

Depending on the `return_imapconf` parameter, it can be a `list` of length `2` containing the `imapconf` object and message ids resulted from search, or a numeric vector containing only message ids.

See Also

Other Size-search operations: `custom_search`, `search_larger_than`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )
```

```
# search
results <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    search_smaller_than(size = 512000) # smaller than 512KB


## End(Not run)
```

---

search_string *Search By String*

Description

Searches for messages containing a string in an specific section or field.

Usage

```
search_string(imapconf, section_or_field, string, negate = FALSE,
  by = "MSN", esearch = FALSE, return_imapconf = TRUE, retries = 2)
```

Arguments

imapconf
: An object of class imapconf generated with configure_imap and with a mbox item added after a mailbox selection with select_mailbox.

section_or_field
: A mandatory character string specifying in which message's Section or Header Field to search for the provided string. For some available options, see section_or_field_options.

string
: A character string specifying the word or expression to search for in messages.

negate
: If TRUE, negates the search and seeks for "NOT search_criterion". Default is FALSE.

by
: Which id (MSN or UID) to use when searching for messages. "MSN" or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. "UID" or unique identifier is always the same during the life cycle of a message. Default is "MSN".

esearch
: A logical. Default is FALSE. If the IMAP server has ESEARCH capability, it can be used to optimize search results. It allows to condense results to message sets to cut down on transmission costs, e.g. 1:5 instead of writing ids individually such as 1,2,3,4,5. It can be used along with buffersize to avoid results stripping. Check if your IMAP server supports SEARCH with list_server_capabilities.

`return_imapconf`
: A logical. If TRUE, the function returns a list of length 2, containing the imapconf object (IMAP settings) and the search results as message ids ("MSN" or "UID"). If FALSE, returns only the message ids as a numeric vector. Default is TRUE.

`retries`
: Number of attempts to connect and execute the command. Default is 2.

Value

Depending on the return_imapconf parameter, it can be a list of length 2 containing the imapconf object and message ids resulted from search, or a numeric vector containing only message ids.

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search
results <- imapconf %>%
    select_mailbox(mbox = "TAM") %>%
    search_string(section_or_field = "FROM",
                  string = "michelle@hotmail.com",
                  negate = TRUE
                 ) # not FROM "michele"


## End(Not run)
```

search_younger_than *Search Within seconds*

Description

Searches for messages within a number of seconds from the server's current time to fetch messages.

Usage

```
search_younger_than(imapconf, seconds, negate = FALSE, by = "MSN",
  flag = NULL, esearch = FALSE, return_imapconf = TRUE,
  retries = 2)
```

Arguments

`imapconf` An object of class `imapconf` generated with configure_imap and with a `mbox` item added after a mailbox selection with select_mailbox.

`seconds` An integer specifying the number of seconds to be used as search criterion.

`negate` If `TRUE`, negates the search and seeks for "NOT search_criterion". Default is `FALSE`.

`by` Which id (`MSN` or `UID`) to use when searching for messages. `"MSN"` or message squence number is a message's relative position to the older message in the mailbox. It may change after deleting or moving messages. For instance, if a message is deleted, message sequence numbers are reordered to fill the gap. `"UID"` or unique identifier is always the same during the life cycle of a message. Default is `"MSN"`.

`flag` Optional parameter that adds a flag filter to the search. Use flag_options to list the common flags used by IMAP servers. Default is `NULL`.

`esearch` A logical. Default is `FALSE`. If the IMAP server has `ESEARCH` capability, it can be used to optimize search results. It allows to condense results to message sets to cut down on transmission costs, e.g. `1:5` instead of writing ids individually such as `1,2,3,4,5`. It can be used along with buffersize to avoid results stripping. Check if your IMAP server supports `SEARCH` with `list_server_capabilities`.

`return_imapconf` A logical. If `TRUE`, the function returns a `list` of length `2`, containing the `imapconf` object (IMAP settings) and the search results as message ids (`"MSN"` or `"UID"`). If `FALSE`, returns only the message ids as a numeric vector. Default is `TRUE`.

`retries` Number of attempts to connect and execute the command. Default is `2`.

Value

Depending on the `return_imapconf` parameter, it can be a `list` of length `2` containing the `imapconf` object and message ids resulted from search, or a numeric vector containing only message ids.

Note

Some SMTP servers, such as Gmail and AOL, may not support WITHIN SEARCH EXTENSION.

See Also

Other Within-search operations: `custom_search`, `search_older_than`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                              username="your_username",
                              password=rstudioapi::askForPassword()
                             )

# search
results <- imapconf %>%
    select_mailbox(mbox = "Smiles GOL") %>%
    search_younger_than(seconds = 3600)


## End(Not run)
```

`section_or_field_options` *Section or Header Fields Options*

Description

This function presents message sections and header fields that can be used for searching and fetching message contents. The header fields, particularly, are only a suggestion and indicate the most common terms. Some messages may not contain one or more of the listed fields. It is also possible that one or more fields not listed here may be used in some messages.

Usage

```
section_or_field_options()
```

Value

A `list` containing common header fields.

Note

A message body is the (full) content of a message. It is simply lines of US-ASCII characters (Freed & Borenstein, 1996; Resnick, 2001). Therefore, if you want to do a search of a string in the whole message, including `HEADER` and `TEXT`, you can specify `section_or_field = "BODY"`.

While header is a part of the message body, indicating sender, date and other information, header fields are specific parts of the header of a message. "Header fields are lines composed of a field name, followed by a colon (":"), which is followed by a field body, and terminated by CRLF" (Resnick, 2001).

A message `TEXT` is the message itself, indicating the content written or produced by the sender.

References

Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, doi: 10.17487/RFC2046, November 1996, https://www.rfc-editor.org/info/rfc2046.

Resnick, P., Ed., "Internet Message Format", RFC 2822, doi: 10.17487/RFC2822, April 2001, https://www.rfc-editor.org/info/rfc2822.

Crocker, D., "STANDARD FOR THE FORMAT OF ARPA INTERNET TEXT MESSAGES", STD 11, RFC 822, doi: 10.17487/RFC0822, August 1982, https://www.rfc-editor.org/info/rfc822.

See Also

Other options: `flag_options`, `metadata_options`

Examples

```
## Not run:

library(mRpostman)
section_or_field_options()


## End(Not run)
```

`select_mailbox` *Mailbox Selection*

Description

Select a mailbox on the server for issuing further commands.

Usage

```
select_mailbox(imapconf, mbox, retries = 2)
```

Arguments

`imapconf`	A object of class `imapconf` generated with configure_imap.
`mbox`	A string containing the name of an existing mailbox on the server.
`retries`	Number of attempts to connect and execute the command. Default is `2`.

Value

The (invisible) `imaconf` object with the selected mailbox added to it.

See Also

Other mailbox commands: examine_mailbox, list_mailboxes, list_server_capabilities, rename_mailbox

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configureIMAP(url="imaps://imap.gmail.com",
                          username="your_gmail_user",
                          password=rstudioapi::askForPassword()
                         )

# Select INBOX
imapconf %>%
  select_mailbox(mbox = "INBOX")


## End(Not run)
```

sent_before *Criteria Helper Functions for Custom Search*

Description

Criteria helper functions to be combined inside a operator helper function as a custom request in custom_search.

Usage

```
sent_before(date_char, negate = FALSE)
```

Arguments

date_char	A character vector with format "DD-Mon-YYYY", e.g. "01-Apr-2019". We opted not to use objects of type "date", since IMAP servers like this not so common date format.
negate	If TRUE, negates the search and seeks for "NOT search_criterion". Default is FALSE.

Value

A search string to be used as a custom_request parameter in custom_search function.

See Also

Other customsearch helper functions: AND, OR, before, flag, larger_than, older_than, on, sent_on, sent_since, since, smaller_than, string, younger_than

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search
result <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    custom_search(custom_request = OR(sent_before(date_char = "17-Apr-2019"),
                                      smaller_than(size = 512000)
                                     ))
# searches for messages sentBefore "17-Apr-2019" OR Smaller Than 512KB.


## End(Not run)
```

---

sent_on *Criteria Helper Functions for Custom Search*

Description

Criteria helper functions to be combined inside a operator helper function as a custom request in custom_search.

Usage

```
sent_on(date_char, negate = FALSE)
```

Arguments

date_char	A character vector with format "DD-Mon-YYYY", e.g. "01-Apr-2019". We opted not to use objects of type "date", since IMAP servers like this not so common date format.
negate	If TRUE, negates the search and seeks for "NOT search_criterion". Default is FALSE.

Value

A search string to be used as a `custom_request` parameter in custom_search function.

See Also

Other customsearch helper functions: `AND`, `OR`, `before`, `flag`, `larger_than`, `older_than`, `on`, `sent_before`, `sent_since`, `since`, `smaller_than`, `string`, `younger_than`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search
result <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    custom_search(custom_request = OR(sentOn(date_char = "17-Apr-2019"),
                                      sentOn(date_char = "17-Jul-2018")
                                     ))
# searches for messages Sent On "17-Apr-2019" OR Sent On "17-Jul-2018".


## End(Not run)
```

`sent_since` *Criteria Helper Functions for Custom Search*

Description

Criteria helper functions to be combined inside a operator helper function as a custom request in custom_search.

Usage

```
sent_since(date_char, negate = FALSE)
```

Arguments

`date_char` A character vector with format "DD-Mon-YYYY", e.g. "01-Apr-2019". We opted not to use objects of type "date", since IMAP servers like this not so common date format.

`negate` If TRUE, negates the search and seeks for "NOT search_criterion". Default is FALSE.

Value

A search string to be used as a custom_request parameter in custom_search function.

See Also

Other customsearch helper functions: AND, OR, before, flag, larger_than, older_than, on, sent_before, sent_on, since, smaller_than, string, younger_than

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search
result <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    custom_search(custom_request = OR(sent_since(date_char = "17-Apr-2019"),
                                      smaller_than(size = 512000)
                                     ))
# searches for messages sentSince "17-Apr-2019" OR Smaller Than 512KB.


## End(Not run)
```

since *Criteria Helper Functions for Custom Search*

Description

Criteria helper functions to be combined inside a operator helper function as a custom request in custom_search.

Usage

```
since(date_char, negate = FALSE)
```

Arguments

date_char A character vector with format "DD-Mon-YYYY", e.g. "01-Apr-2019". We opted not to use objects of type "date", since IMAP servers like this not so common date format.

negate If TRUE, negates the search and seeks for "NOT search_criterion". Default is FALSE.

Value

A search string to be used as a custom_request parameter in custom_search function.

See Also

Other customsearch helper functions: AND, OR, before, flag, larger_than, older_than, on, sent_before, sent_on, sent_since, smaller_than, string, younger_than

Examples

```

## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search
result <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    custom_search(custom_request = AND(since(date_char = "17-Apr-2019"),
                                       smaller_than(size = 512000)
                                      ))
# searches for messages Since "17-Apr-2019" AND Smaller Than  512KB.


## End(Not run)
```

smaller_than *Criteria Helper Functions for Custom Search*

Description

Criteria helper functions to be combined inside a operator helper function as a custom request in custom_search.

Usage

```
smaller_than(size, negate = FALSE)
```

Arguments

size	An integer specifying the size (in number of characters) of message to be used as search criterion.
negate	If TRUE, negates the search and seeks for "NOT search_criterion". Default is FALSE.

Value

A search string to be used as a custom_request parameter in custom_search function.

See Also

Other customsearch helper functions: AND, OR, before, flag, larger_than, older_than, on, sent_before, sent_on, sent_since, since, string, younger_than

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search
result <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    custom_search(custom_request =
                 OR(string(
                       section_or_field = "from", string = "allanvcq@gmail.com"),
                    smaller_than(
                       size = 512000)
                   )
                 )
# searches for messages containing the string "allanvcq@gmail.com" in the
# field "FROM" OR those Smaller Than 512KB.


## End(Not run)
```

---

string *Criteria Helper Functions for Custom Search*

Description

Criteria helper functions to be combined inside a operator helper function as a custom request in custom_search.

Usage

```
string(section_or_field, string, negate = FALSE)
```

Arguments

`section_or_field`
A mandatory character string specifying in which messages's Section or Header Field to search for the provided string. For some available options, see section_or_field_options.

`string` A character string specifying the word or expression to search for in messages.

`negate` If `TRUE`, negates the search and seeks for "NOT search_criterion". Default is `FALSE`.

Value

A search string to be used as a `custom_request` parameter in custom_search function.

See Also

Other customsearch helper functions: `AND`, `OR`, `before`, `flag`, `larger_than`, `older_than`, `on`, `sent_before`, `sent_on`, `sent_since`, `since`, `smaller_than`, `younger_than`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search
result <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    custom_search(custom_request =
                 OR(string(
                       section_or_field = "from", string = "allanvcq@gmail.com"),
                    string(
                       section_or_field = "from", string = "allanvcq@yahoo.com")
                   )
                 )
# searches for messages containing the string "allanvcq@gmail.com" OR
# "allanvcq@yahoo.com" in the "FROM" field.


## End(Not run)
```

younger_than *Criteria Helper Functions for Custom Search*

Description

Criteria helper functions to be combined inside a operator helper function as a custom request in custom_search.

Usage

```
younger_than(seconds, negate = FALSE)
```

Arguments

`seconds`	An integer specifying the number of seconds to be used as search criterion.
`negate`	If `TRUE`, negates the search and seeks for "NOT search_criterion". Default is `FALSE`.

Value

A search string to be used as a `custom_request` parameter in custom_search function.

See Also

Other customsearch helper functions: `AND`, `OR`, `before`, `flag`, `larger_than`, `older_than`, `on`, `sent_before`, `sent_on`, `sent_since`, `since`, `smaller_than`, `string`

Examples

```
## Not run:

# configure IMAP
library(mRpostman)
imapconf <- configure_imap(url="imaps://your.imap.server.com",
                           username="your_username",
                           password=rstudioapi::askForPassword()
                          )

# search
result <- imapconf %>%
    select_mailbox(mbox = "INBOX") %>%
    custom_search(custom_request =
                 OR(smaller_than(
                       size = 512000),
                    younger_than(
                       seconds = 3600)
                   )
                 )
# searches for messages that are Smaller Than 512 KB OR Younger Than 3600
# seconds (or one hour).


## End(Not run)
```

Index