

Package ‘emayili’

September 6, 2021

Type Package

Title Send Email Messages

Version 0.4.18

Description A light, simple tool for sending emails with minimal dependencies.

URL <https://datawookie.github.io/emayili/>

BugReports <https://github.com/datawookie/emayili/issues>

License GPL-3

Imports base64enc, curl (>= 4.0), dplyr, glue, httr, magrittr, mime, purrr, stringi, stringr, tibble, tidyr, vctrs

Suggests here, testthat (>= 2.1.0), roxygen2, showtext

Encoding UTF-8

RoxygenNote 7.1.1

NeedsCompilation no

Author Andrew B. Collier [aut, cre],
Matt Dennis [ctb],
Antoine Bichat [ctb] (<<https://orcid.org/0000-0001-6599-7081>>),
Daniel Fahey [ctb],
Johann R. Kleinbub [ctb],
Panagiotis Moulos [ctb]

Maintainer Andrew B. Collier <andrew.b.collier@gmail.com>

Repository CRAN

Date/Publication 2021-09-06 11:40:02 UTC

R topics documented:

address	2
as.address	3
as.character.envelope	4
as.character.vctrs_address	5
attachment	5

bcc	6
cc	7
compare	8
compliant	8
display	9
domain	9
envelope	10
format.mime	11
format.vctrs_address	11
from	12
header	13
html	13
local	14
mime	15
new_address	15
parameter_value_encode	16
parties	17
print.envelope	17
print.vctrs_address	18
qp_decode	18
qp_encode	19
raw	19
reply	20
sender	20
server	21
subject	23
text	23
to	25
vec_ptype_abbr.vctrs_address	25
vec_ptype_full.vctrs_address	26
Index	27

address

Email Address

Description

Create an address object which represents an email address.

Usage

```
address(email = NA, display = NA, local = NA, domain = NA, normalise = TRUE)
```

Arguments

email	Email address.
display	Display name.
local	Local part of email address.
domain	Domain part of email address.
normalise	Whether to try to normalise address to RFC-5321 requirements.

Details

Implemented as an **S3 vector class**.

Value

An address object, representing an email address.

Examples

```
address("gerry@gmail.com")
address("gerry@gmail.com", "Gerald")
address(
  c("gerry@gmail.com", "alice@yahoo.com", "jim@aol.com"),
  c("Gerald", "Alice", NA)
)
```

as.address	<i>Create an address object</i>
------------	---------------------------------

Description

Create an address object

Usage

```
as.address(addr)
```

Arguments

addr	An email address.
------	-------------------

Value

An address object.

Examples

```
as.address("gerry@gmail.com")
as.address("Gerald <gerry@gmail.com>")
as.address(c("Gerald <gerry@gmail.com>", "alice@yahoo.com", "jim@aol.com"))
```

as.character.envelope *Create formatted message.*

Description

Accepts a message object and formats it as a MIME document.

Usage

```
## S3 method for class 'envelope'  
as.character(x, ...)
```

Arguments

x A message object.
... Further arguments passed to or from other methods.

Value

A formatted message object.

MIME Multipart Types

There are a number of options for multipart messages:

- multipart/mixed — Used for sending content with multiple independent parts either inline or as attachments. Each part can have different Content-Type.
- multipart/alternative — Used when each part of the message is an "alternative" version of the same content. The order of the parts is important: preferred and/or more complex formats should be found towards the end.
Example: A message with both plain text and HTML versions.
- multipart/digest — Used to send multiple plain text messages.
- multipart/related — Used when each part of the the message represents a component of the complete message.
Example: A web page with images.
- multipart/signed — Used when a message has a digital signature attached.
- multipart/encrypted — Used for a message with encrypted content.

A nice illustration of how some of these relate can be found at <https://stackoverflow.com/a/40420648/633961>.

`as.character.vctrs_address`*Convert address object to character*

Description

Convert address object to character

Usage

```
## S3 method for class 'vctrs_address'  
as.character(x, ...)
```

Arguments

`x` A vector of address objects.
`...` Further arguments passed to or from other methods.

Value

A character vector.

`attachment`*Add attachments to a message object*

Description

Add attachments to a message object

Usage

```
attachment(  
  msg,  
  path,  
  name = NA,  
  type = NA,  
  cid = NA,  
  disposition = "attachment"  
)
```

Arguments

msg	A message object.
path	Path to file.
name	Name to be used for attachment (defaults to base name of path).
type	MIME type or <i>NA</i> , which will result in a guess based on file extension.
cid	Content-ID or <i>NA</i> .
disposition	Should the content be displayed inline or as an attachment? Valid options are "inline" and "attachment". If set to <i>NA</i> then will guess appropriate value.

Value

A message object.

Examples

```
library(magrittr)

path_mtcars <- tempfile(fileext = ".csv")
path_scatter <- tempfile(fileext = ".png")
path_cats <- system.file("cats.jpg", package = "emayili")

write.csv(mtcars, path_mtcars)

png(path_scatter)
plot(1:10)
dev.off()

msg <- envelope() %>%
  attachment(path_mtcars) %>%
  # This attachment will have file name "cats.jpg".
  attachment(path_cats, name = "cats.jpg", type = "image/jpeg") %>%
  attachment(path_scatter, cid = "scatter")

file.remove(path_scatter, path_mtcars)
```

bcc

Add Bcc field to message

Description

Add Bcc field to message

Usage

```
bcc(msg, ...)
```

Arguments

msg A message object.
... Email addresses.

Value

A message object.

See Also

[to](#), [cc](#), [from](#), [sender](#), [reply](#) and [subject](#)

Examples

```
msg <- envelope()  
bcc(msg, "bob@gmail.com", "alice@yahoo.com")  
bcc(msg, c("bob@gmail.com", "alice@yahoo.com"))
```

cc *Add Cc field to message*

Description

Add Cc field to message

Usage

```
cc(msg, ...)
```

Arguments

msg A message object.
... Email addresses.

Value

A message object.

See Also

[to](#), [bcc](#), [from](#), [sender](#), [reply](#) and [subject](#)

Examples

```
msg <- envelope()  
cc(msg, "bob@gmail.com", "alice@yahoo.com")  
cc(msg, c("bob@gmail.com", "alice@yahoo.com"))
```

compare	<i>Compare vectors</i>
---------	------------------------

Description

Returns TRUE wherever elements are the same (including NA), and FALSE everywhere else.

Usage

```
compare(lhs, rhs)
```

Arguments

lhs	LHS of operation.
rhs	RHS of operation.

Value

A Boolean value.

compliant	<i>Tests whether an email address is syntactically correct</i>
-----------	--

Description

Checks whether an email address conforms to the **syntax rules**.

Usage

```
compliant(addr)
```

Arguments

addr	An email address.
------	-------------------

Details

An email address may take either of the following forms:

- local@domain or
- Display Name <local@domain>.

Value

A Boolean.

Examples

```
compliant("alice@example.com")  
compliant("alice?example.com")
```

display *Extract display name*

Description

Extracts the display name from an email address.

Usage

```
display(addr)
```

Arguments

addr An address object.

Value

The display name or NA.

Examples

```
gerry <- as.address("Gerald <gerry@gmail.com>")  
display(gerry)
```

domain *Extract domain of email address*

Description

Extract domain of email address

Usage

```
domain(addr)
```

Arguments

addr An address object.

Value

A character vector.

Examples

```
domain("alice@example.com")
```

envelope

Create a message.

Description

Create a message.

Usage

```
envelope(  
  to = NULL,  
  from = NULL,  
  cc = NULL,  
  bcc = NULL,  
  reply = NULL,  
  subject = NULL,  
  text = NULL,  
  html = NULL  
)
```

Arguments

to	See to()
from	See from()
cc	See cc()
bcc	See bcc()
reply	See reply()
subject	See subject()
text	See text()
html	See html()

Value

A message object.

See Also

[subject](#), [from](#), [to](#), [cc](#), [bcc](#) and [reply](#)

Examples

```
# Create an (empty) message object.
msg <- envelope()

# Create a complete message object.
envelope(
  to = "bob@gmail.com",
  from = "craig@gmail.com",
  cc = "alex@gmail.com",
  bcc = "shannon@gmail.com",
  reply = "craig@yahoo.com",
  subject = "Hiya!",
  text = "Hi Bob, how are you?"
)
```

format.mime

Format the header of a MIME object.

Description

The header must conform to the SMTP Protocol (<https://tools.ietf.org/html/rfc821>). Specifically, lines should be terminated by CRLF (not just LF). The “name” and “filename” fields can contain any possible UTF-8 characters as they are [parameter value encoded](#).

Usage

```
## S3 method for class 'mime'
format(msg)
```

Arguments

msg A message object.

Value

A formatted header string.

format.vctrs_address

Encode email addresses in a common format

Description

Encode email addresses in a common format

Usage

```
## S3 method for class 'vctrs_address'  
format(x, ...)
```

Arguments

`x` A vector of address objects.
`...` Further arguments passed to or from other methods.

Value

A character vector.

from	<i>Add From field to message</i>
------	----------------------------------

Description

Add From field to message

Usage

```
from(msg, from = NULL)
```

Arguments

`msg` A message object.
`from` Email address.

Value

A message object.

See Also

[to](#), [cc](#), [bcc](#), [sender](#), [reply](#) and [subject](#)

Examples

```
msg <- envelope()  
from(msg, "craig@gmail.com")
```

header	<i>Create formatted header.</i>
--------	---------------------------------

Description

Create formatted header.

Usage

```
header(msg)
```

Arguments

msg	A message object.
-----	-------------------

Value

A message header.

html	<i>Add an HTML body to a message object.</i>
------	--

Description

Add an HTML body to a message object.

Usage

```
html(  
    msg,  
    content,  
    disposition = "inline",  
    charset = "utf-8",  
    encoding = "quoted-printable"  
)
```

Arguments

msg	A message object.
content	A string of message content or the path to a file containing HTML.
disposition	How content is presented (Content-Disposition).
charset	How content is encoded.
encoding	How content is transformed to ASCII (Content-Transfer-Encoding).

Value

A message object.

See Also

[text](#)

Examples

```
library(magrittr)

# Inline HTML message.
msg <- envelope() %>% html("<b>Hello!</b>")

# Read HTML message from a file.
msg <- envelope() %>% html("message.html")
```

local

Extract local part of email address

Description

Extract local part of email address

Usage

```
local(addr)
```

Arguments

addr An address object.

Value

A character vector.

Examples

```
local("alice@example.com")
```

mime	<i>Create a MIME (Multipurpose Internet Mail Extensions) object.</i>
------	--

Description

Create a MIME (Multipurpose Internet Mail Extensions) object.

Usage

```
mime(content_type, content_disposition, charset, encoding, cid = NA, ...)
```

Arguments

content_type	The MIME type of the content.
content_disposition	Should the content be displayed inline or as an attachment?
charset	How to interpret the characters in the content. Most often either UTF-8 or ISO-8859-1.
encoding	How to encode binary data to ASCII.
cid	An optional Content-Id.
...	Other arguments.

Value

A MIME object.

new_address	<i>Helper function for creating address objects</i>
-------------	---

Description

Helper function for creating address objects

Usage

```
new_address(  
  email = character(),  
  display = character(),  
  local = character(),  
  domain = character(),  
  normalise = TRUE  
)
```

Arguments

email	Email address.
display	Display name.
local	Local part of email address.
domain	Domain part of email address.
normalise	Whether to try to normalise address to RFC-5321 requirements.

Value

An address object, representing an email address.

parameter_value_encode

Encode parameter values for MIME headers

Description

Formats UTF-8 parameter values for the use in MIME headers. Non-ASCII characters, control characters, and some special characters have to be specially encoded according to the [RFC2231](#) specification.

Usage

parameter_value_encode(x)

Arguments

x character(1). UTF-8 string to format.

Value

character(1). String to put right after the parameter name in a MIME header. The equal sign and possible quotation marks are included. For example "I'm not quoted.csv" turns to "=\"I'm not quoted.csv\"" while "\"I'm quoted\".csv" results in "*=utf-8''%22I'm%20quoted%22.csv".

Examples

```
emayili::parameter_value_encode("I'm not quoted.csv")
emayili::parameter_value_encode("\"I'm quoted\".csv")
```

parties	<i>Extract sender and recipient(s)</i>
---------	--

Description

Extract sender and recipient(s)

Usage

```
parties(msg)
```

Arguments

msg	A message object.
-----	-------------------

Value

A tibble.

Examples

```
email <- envelope() %>%
  from("Gerald <gerald@gmail.com>") %>%
  to(c("bob@gmail.com", "alice@yahoo.com")) %>%
  cc("Craig <craig@gmail.com>") %>%
  bcc(" Erin <erin@yahoo.co.uk >")

parties(email)
```

print.envelope	<i>Print a message object</i>
----------------	-------------------------------

Description

The message body will be printed if details is TRUE or if the envelope_details option is TRUE.

Usage

```
## S3 method for class 'envelope'
print(x, details = NA, ...)
```

Arguments

x	A message object.
details	Whether or not to display full message content.
...	Any other arguments (for consistency of generic function).

Examples

```
msg <- envelope() %>% text("Hello, World!")

print(msg)
print(msg, details = TRUE)

options(envelope_details = TRUE)
print(msg)
```

```
print.vctrs_address   Print an address object
```

Description

Print an address object

Usage

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

Arguments

x	An address object.
...	Further arguments passed to or from other methods.

Examples

```
gerry <- as.address("gerry@gmail.com")
print(gerry)
```

```
qp_decode             Decode a quoted-printable string.
```

Description

Decode a quoted-printable string.

Usage

```
qp_decode(x)
```

Arguments

x	A vector of strings.
---	----------------------

Value

A vector of decoded strings.

qp_encode	<i>Encode a string to quoted-printable.</i>
-----------	---

Description

Encode a string to quoted-printable.

Usage

```
qp_encode(x)
```

Arguments

x A vector of strings.

Value

A vector of encoded strings.

raw	<i>Extract raw email address</i>
-----	----------------------------------

Description

Strips the display name off an email address (if present).

Usage

```
raw(addr)
```

Arguments

addr An address object.

Value

A raw email address.

Examples

```
gerry <- as.address("Gerald <gerry@gmail.com>")
raw(gerry)
```

reply	<i>Add Reply-To field to message</i>
-------	--------------------------------------

Description

Add Reply-To field to message

Usage

```
reply(msg, reply_to = NULL)
```

Arguments

msg	A message object.
reply_to	Email address.

Value

A message object.

See Also

[to](#), [cc](#), [bcc](#), [from](#), [sender](#) and [subject](#)

Examples

```
msg <- envelope()  
reply(msg, "gerry@gmail.com")
```

sender	<i>Add Sender (on behalf of) field to message</i>
--------	---

Description

Add Sender (on behalf of) field to message

Usage

```
sender(msg, sender = NULL)
```

Arguments

msg	A message object.
sender	Email address.

Value

A message object.

See Also

[to](#), [cc](#), [bcc](#), [from](#), [reply](#) and [subject](#)

Examples

```
msg <- envelope()
sender(msg, "on_behalf_of@gmail.com")
```

server

Create a SMTP server object.

Description

Create a SMTP server object.

Usage

```
server(
  host,
  port = 25,
  username = NULL,
  password = NULL,
  insecure = FALSE,
  reuse = TRUE,
  helo = NA,
  pause_base = 1,
  max_times = 5,
  ...
)
```

Arguments

host	DNS name or IP address of the SMTP server.
port	Port that the SMTP server is listening on.
username	Username for SMTP server.
password	Password for SMTP server.
insecure	Whether to ignore SSL issues.
reuse	Whether the connection to the SMTP server should be left open for reuse.
helo	The HELO domain name of the sending host. If left as NA then will use local host name.
pause_base	Base delay (in seconds) for exponential backoff. See rate_backoff .

max_times Maximum number of times to retry.
... Additional curl options. See `curl::curl_options()` for a list of supported options.

Value

A function which is used to send messages to the server.

Examples

```
library(magrittr)

# Set parameters for SMTP server (with username and password)
smtp <- server(host = "smtp.gmail.com",
               port = 465,
               username = "bob@gmail.com",
               password = "bd40ef6d4a9413de9c1318a65cbae5d7")

# Set parameters for a (fake) testing SMTP server.
#
# More information about this service can be found at https://www.smtpbucket.com/.
#
smtp <- server(host = "mail.smtpbucket.com",
               port = 8025)

# Create a message
msg <- envelope() %>%
  from("bob@gmail.com") %>%
  to("alice@yahoo.com")

# Send message (verbose output from interactions with server)
## Not run:
smtp(msg, verbose = TRUE)

## End(Not run)

# To confirm that the message was sent, go to https://www.smtpbucket.com/ then:
#
# - fill in "bob@gmail.com" for the Sender field and
# - fill in "alice@yahoo.com" for the Recipient field then
# - press the Search button.

# With explicit HELO domain.
#
smtp <- server(host = "mail.example.com",
               helo = "client.example.com")
```

subject	<i>Add or query subject of message.</i>
---------	---

Description

Add or query subject of message.

Usage

```
subject(msg, subject = NULL)
```

Arguments

msg	A message object.
subject	A subject for the message.

Value

A message object or the subject of the message object (if subject is NULL).

See Also

[to](#), [from](#), [cc](#), [bcc](#) and [reply](#)

Examples

```
library(magrittr)

# Create a message and set the subject
msg <- envelope() %>% subject("Updated report")

# Retrieve the subject for a message
subject(msg)
```

text	<i>Add a text body to a message.</i>
------	--------------------------------------

Description

Uses `glue::glue()` to evaluate expressions enclosed in brackets as R code.

Usage

```
text(  
  msg,  
  content,  
  disposition = "inline",  
  charset = "utf-8",  
  encoding = "7bit",  
  interpolate = TRUE,  
  .envir = NULL  
)
```

Arguments

<code>msg</code>	A message object.
<code>content</code>	A string of message content.
<code>disposition</code>	How content is presented (Content-Disposition).
<code>charset</code>	How content is encoded.
<code>encoding</code>	How content is transformed to ASCII (Content-Transfer-Encoding).
<code>interpolate</code>	Whether to interpolate using glue.
<code>.envir</code>	Environment used for glue interpolation. Defaults to <code>parent.frame()</code> .

Value

A message object.

See Also

[html](#)

Examples

```
library(magrittr)  
  
msg <- envelope() %>% text("Hello!")  
  
# Using {glue} interpolation.  
#  
name <- "Alice"  
msg <- envelope() %>% text("Hello {name}.")  
  
print(msg, details = TRUE)  
  
# Disable {glue} interpolation.  
#  
msg <- envelope() %>% text("This is a set: {1, 2, 3}.", interpolate = FALSE)
```

to *Add To field to message*

Description

Add To field to message

Usage

```
to(msg, ...)
```

Arguments

msg	A message object.
...	Email addresses.

Value

A message object.

See Also

[cc](#), [bcc](#), [from](#), [sender](#), [reply](#) and [subject](#)

Examples

```
msg <- envelope()  
to(msg, "bob@gmail.com", "alice@yahoo.com")  
to(msg, c("bob@gmail.com", "alice@yahoo.com"))
```

vec_ptype_abbr.vctrs_address

Display abbreviated type of vector

Description

Display abbreviated type of vector

Usage

```
vec_ptype_abbr.vctrs_address(x, ...)
```

Arguments

x	A vector.
...	These dots are for future extensions and must be empty.

`vec_ptype_full.vctrs_address`*Display full type of vector*

Description

Display full type of vector

Usage

```
vec_ptype_full.vctrs_address(x, ...)
```

Arguments

<code>x</code>	A vector.
<code>...</code>	These dots are for future extensions and must be empty.

Index

address, [2](#)
as.address, [3](#)
as.character.envelope, [4](#)
as.character.vctr_address, [5](#)
attachment, [5](#)

bcc, [6](#), [7](#), [10](#), [12](#), [20](#), [21](#), [23](#), [25](#)

cc, [7](#), [7](#), [10](#), [12](#), [20](#), [21](#), [23](#), [25](#)
compare, [8](#)
compliant, [8](#)

display, [9](#)
domain, [9](#)

envelope, [10](#)

format.mime, [11](#)
format.vctr_address, [11](#)
from, [7](#), [10](#), [12](#), [20](#), [21](#), [23](#), [25](#)

header, [13](#)
html, [13](#), [24](#)

local, [14](#)

mime, [15](#)

new_address, [15](#)

parameter value encoded, [11](#)
parameter_value_encode, [16](#)
parties, [17](#)
print.envelope, [17](#)
print.vctr_address, [18](#)

qp_decode, [18](#)
qp_encode, [19](#)

rate_backoff, [21](#)
raw, [19](#)
reply, [7](#), [10](#), [12](#), [20](#), [21](#), [23](#), [25](#)

sender, [7](#), [12](#), [20](#), [20](#), [25](#)
server, [21](#)
subject, [7](#), [10](#), [12](#), [20](#), [21](#), [23](#), [25](#)

text, [14](#), [23](#)
to, [7](#), [10](#), [12](#), [20](#), [21](#), [23](#), [25](#)

vec_ptype_abbr.vctr_address, [25](#)
vec_ptype_full.vctr_address, [26](#)