Package ‘emayili’

June 3, 2020

Type Package
Title Send Email Messages
Version 0.4.0
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, glue, httr, magrittr, mime, stringi
Suggests testthat (>= 2.1.0), roxygen2
Encoding UTF-8
LazyData true
RoxygenNote 7.1.0
NeedsCompilation no
Author Andrew B. Collier [aut, cre],
    Gerard Walsh [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@exegetic.biz>
Repository CRAN
Date/Publication 2020-06-03 14:20:03 UTC

R topics documented:

  attachment .......................................................... 2
  bcc ................................................................. 3
  cc ................................................................. 4
  envelope ......................................................... 4
  format.mime ..................................................... 5
attachment

Add attachments to a message object

Description

Add attachments to a message object

Usage

attachment(msg, path, type = NA, cid = NA, disposition = NA)

Arguments

msg A message object.
path Path to file.
type MIME type or NA, which will result in a guess based on file extension.
cid Content-ID or NA.
disposition Should the content be displayed inline or as an attachment?

Value

A message object.

Examples

library(magrittr)

path_mtcars <- tempfile(fileext = "csv")
path_cats <- tempfile(fileext = "jpg")
path_scatter <- tempfile(fileext = "png")

write.csv(mtcars, path_mtcars)

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

msg <- envelope() %>%
  attachment(path_mtcars) %>%
  attachment(path_cats, type = "image/jpeg") %>%
  attachment(path_scatter, cid = "scatter")

file.remove(path_cats, 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**

`subject`, `from`, `to`, `cc` and `reply`

**Examples**

```r
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
subject, from, to, bcc and reply

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

envelope
Create a message.

Description
Create a message.

Usage
envelope()

Value
A message object.
See Also

subject, from, to, cc, bcc and reply

Examples

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

format.mime

Format the header of a MIME object.

Description


Usage

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

Arguments

msg A message object.

Value

A formatted header string.

from

Add From field to message

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
subject, to, cc, bcc and reply

Examples

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

---

**header**

*Create formatted header.*

### Description
Create formatted header.

### Usage

```r
header(msg)
```

### Arguments

- `msg` A message object.

### Value
A message header.

---

**html**

*Add an HTML body to a message object.*

### Description
Add an HTML body to a message object.

### Usage

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

Arguments

- `msg` : A message object.
- `content` : A string of message content.
- `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

```r
library(magrittr)

msg <- envelope() %>% html("<b>Hello!</b>"
```

message  Create formatted message.

Description

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

Usage

```r
message(msg)
```

Arguments

- `msg` : A message object.

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 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 http://stackoverflow.com/a/40420648/633961.

---

**mime**

Create a MIME (Multipurpose Internet Mail Extensions) object.

**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.
print.envelope  

**Print a message object**

**Description**

Print a message object

**Usage**

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

**Arguments**

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

**qp_decode**  

**Decode a quoted-printable string.**

**Description**

Decode a quoted-printable string.

**Usage**

```r
qp_decode(x)
```

**Arguments**

- `x`  
  A vector of strings.

**Value**

A vector of decoded strings.
qp_encode  
*Encode a string to quoted-printable.*

**Description**
Encode a string to quoted-printable.

**Usage**
qp_encode(x)

**Arguments**
- x  
  A vector of strings.

**Value**
A vector of encoded strings.

---

reply  
*Add Reply-To field to message*

**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**
subject, from, to, cc and bcc

**Examples**
msg <- envelope()
reply(msg, "gerry@gmail.com")
server

Create a SMTP server object.

Description

Create a SMTP server object.

Usage

```r
server(
  host,
  port = 25,
  username = NULL,
  password = NULL,
  insecure = FALSE,
  reuse = TRUE,
  ...
)
```

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.
- **...** 
  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

```r
library(magrittr)

# Set parameters for SMTP server (with username and password)
smtpl <- server(host = "smtp.gmail.com",
                 port = 465,
                 username = "bob@gmail.com",
                 password = "bd40ef6d4a9413de9c1318a65c8a5d7")

# Set parameters for a (fake) testing SMTP server.
#
```
subject

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.

subject

Add or query subject of message.

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

```r
library(magrittr)

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

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

---

**text**

*Add a text body to a message.*

**Description**

Add a text body to a message.

**Usage**

```r
text(
  msg,
  content,
  disposition = "inline",
  charset = "utf-8",
  encoding = "7bit"
)
```

**Arguments**

- `msg`: A message object.
- `content`: A string of message content.
- `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**

- `html`

**Examples**

```r
library(magrittr)

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

Add To field to message

Usage

to(msg, ...)

Arguments

- `msg` A message object.
- `...` Email addresses.

Value

A message object.

See Also

subject, from, cc, bcc and reply

Examples

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

attachment, 2
bcc, 3, 4–6, 10, 12, 14
cc, 3, 4, 5, 6, 10, 12, 14
envelope, 4
format.mime, 5
from, 3–5, 5, 10, 12, 14
header, 6
html, 6, 13
message, 7
mime, 8
print.envelope, 9
qp_decode, 9
qp_encode, 10
reply, 3–6, 10, 12, 14
server, 11
subject, 3–6, 10, 12, 14
text, 7, 13
to, 3–6, 10, 12, 14