

Package ‘mailR’

May 31, 2014

Type Package

Title Generate mails and send them via sendmail

Author Meik Michalke

Maintainer Meik Michalke <meik.michalke@hhu.de>

Depends R (>= 2.9.0),methods

Description You can use this package to generate the console commands to send mail with the sendmail command line tool. Digital encryption via OpenPGP is also supported, signatures don't work yet. You need a working setup of the required tools, namely postfix/sendmail, qprint and GnuPG, for this package to be really useful. It was originally inspired by the need of the C3S project (<https://C3S.cc>) for a mail interface to R. Note that mailR is in its early stages and most likely not ready for professional use!

License GPL (>= 3)

Encoding UTF-8

LazyLoad yes

URL <http://reaktanz.de/?c=hacking&s=mailR>

Version 0.02-1

Date 2014-05-31

Collate '00_internal_functions.R' '01_class_01_email.R' '02_method_01_show.R' 'genMail.R' 'mail.R' 'mailR-package.R' 'mailx.R' 'qprint.R'

R topics documented:

mailR-package	2
email,-class	2
genMail	3
mail	4

mail	6
mailx	7
qprint	8
show,email-method	8

Index 10

mailR-package *The mailR Package*

Description

Generate mails and send them via sendmail.

Details

```

Package:    mailR
Type:       Package
Version:    0.02-1
Date:       2014-05-31
Depends:    R (>= 2.9.0),methods
Encoding:   UTF-8
License:    GPL (>= 3)
LazyLoad:   yes
URL:        http://reaktanz.de/?c=hacking&s=mailR

```

You can use this package to generate the console commands to send mail with the sendmail command line tool. Digital encryption via OpenPGP is also supported, signatures don't work yet. You need a working setup of the required tools, namely postfix/sendmail, qprint and GnuPG, for this package to be really useful. It was originally inspired by the need of the C3S project (<https://C3S.cc>) for a mail interface to R. Note that mailR is in its early stages and most likely not ready for professional use!

Author(s)

Meik Michalke

email,-class *Class email*

Description

This class is used as a container for e-mails.

Details

You should use the constructor function [genMail](#) to generate email objects.

Slots

`body` Character string, the actual mail body.

`subject` Character string, subject of e-mail.

`to` Character string, mail address of the recipient.

`from` Character string, mail address of the sender.

`bcc` Character string, mail address for blind carbon copies.

`replyto` Character string, mail address to be used for replies.

`OpenPGPKeyFrom` Character string, OpenPGP key ID of the sender, used for signing the message.

`OpenPGPKeyTo` Character string, OpenPGP key ID of the recipient, used for encrypting the message.

`boundaryEncrypted` Character string, MIME boundary used for the encrypted parts. If not set, a random boundary will be generated automatically.

`boundaryPlain` Character string, MIME boundary used inside the plain message. If not set, a random boundary will be generated automatically.

`charset` Character string, character encoding of the plain message body. Defaults to "us-ascii".

`ContentTransferEncoding` Character string, set the transfer encoding for the message body, e.g., "7bit" or "quoted-printable" (default).

See Also

[genMail](#)

genMail

Generate objects of class email

Description

Constructor function to generate email messages to be used by [mail](#).

Usage

```
genMail(body, subject, to, from = "", bcc = "", replyto = "",
        OpenPGPKeyFrom = "", OpenPGPKeyTo = "",
        boundaryEncrypted = MIMEboundary(), boundaryPlain = MIMEboundary(),
        charset = "us-ascii", ContentTransferEncoding = "quoted-printable",
        encodeQP = TRUE)
```

Arguments

body	Character string, the actual mail body.
subject	Character string, subject of e-mail.
to	Character string, mail address of the recipient.
from	Character string, mail address of the sender.
bcc	Character string, mail address for blind carbon copies.
replyto	Character string, mail address to be used for replies.
OpenPGPKeyFrom	Character string, OpenPGP key ID of the sender, used for signing the message.
OpenPGPKeyTo	Character string, OpenPGP key ID of the recipient, used for encrypting the message.
boundaryEncrypted	Character string, MIME boundary used for the encrypted parts. If not set, a random boundary will be generated automatically.
boundaryPlain	Character string, MIME boundary used inside the plain message. If not set, a random boundary will be generated automatically.
charset	Character string, character encoding of the plain message body. Defaults to "us-ascii".
ContentTransferEncoding	Character string, set the transfer encoding for the message body, e.g., "7bit" or "quoted-printable" (default).
encodeQP	Logical, whether the message body should automatically be encoded quoted-printable, by calling <code>qprint</code> with default parameters.

Value

An object of class `email`.

See Also

`mail`, `qprint`, `email`

mail

Send mail via sendmail

Description

This method generates a ready-to-use console command to send mails using the sendmail command line tool.

Usage

```
mail(obj, options = "-t -oi", messageOnly = FALSE, encrypt = TRUE,
     sign = TRUE, keyring = NULL, micalg = "pgp-sha1")

## S4 method for signature 'email'
mail(obj, options = "-t -oi", messageOnly = FALSE,
     encrypt = TRUE, sign = TRUE, keyring = NULL, micalg = "pgp-sha1")
```

Arguments

obj	An object of class email.
options	Character string, options which will be passed to sendmail.
messageOnly	Logical, if FALSE, omits the sendmail command line part and prints only the mail headers and message
encrypt	Logical, if TRUE tries to encrypt the message with GnuPG. Only effective if the email object also contains at least one key ID in OpenPGPKeyTo.
sign	Logical, if TRUE tries to sign the message with GnuPG. Only effective if the email object also contains a key ID in OpenPGPKeyFrom.
keyring	Character string, optional path to an additional keyring to look for needed keys.
micalg	Character string, possibility to override the default "micalg" argument for signed messages if another algorithm was used.

Details

You need to have sendmail installed and set up correctly for the mail delivery to work. Setup instructions on sendmail are beyond the scope of this documentation.

Value

A character string, which can be fed to system() to actually send the mail.

Encryption/Signatures

To be able to use the signature and encryption features, you must have a working GnuPG setup. All keys called for must be present in your keyrings. If you set encryption=TRUE and provide a missing or invalid key ID in the email object, the mail will not be generated and an error is given.

See Also

[genMail](#) to generate message objects, [email](#)

Examples

```
exampleMail <- genMail(
  to="foo@example.com",
  subject="this is a test!",
  body="hi,\n\ni'm just testing.\n\nbye\n",
  from="me@example.com")
```

```
# inspect the generated commands
cat(mail(exampleMail))
## Not run:
# evaluate commands, i.e., send the mail
system(mail(exampleMail), intern=TRUE)

## End(Not run)
```

mail

Send mail via mailx

Description

This simple function generates a ready-to-use console command to send mails using the mailx command line tool.

Usage

```
mail(to, subject, message, from = NULL, bcc = NULL,
      replyto = NULL)
```

Arguments

to	Character string, mail address of the recipient.
subject	Character string, subject of the message.
message	Character string, the actual mail body.
from	Character string, mail address of the sender.
bcc	Character string, mail address for blind carbon copies.
replyto	Character string, mail address to be used for replies.

Details

You need to have mailx installed and set up correctly for the mail delivery to work. Setup instructions on mailx are beyond the scope of this documentation.

Value

A character string, which can be fed to system() to actually send the mail.

Note

Try the heirloom-mailx package instead of bsd-mailx on Debian based distros.

Examples

```
exampleMail <- mail("foo@example.com", "this is a test!",
  "hi,\n\ni'm just testing.\n\nbye\n", "me@example.com")
# inspect the generated commands
cat(exampleMail)
## Not run:
# evaluate commands, i.e., send the mail
system(exampleMail, intern=TRUE)

## End(Not run)
```

mailx

Send mail via mailx

Description

This simple function generates a ready-to-use console command to send mails using the mailx command line tool.

Usage

```
mailx(to, subject, message, from = NULL, bcc = NULL, replyto = NULL)
```

Arguments

to	Character string, mail address of the recipient.
subject	Character string, subject of the message.
message	Character string, the actual mail body.
from	Character string, mail address of the sender.
bcc	Character string, mail address for blind carbon copies.
replyto	Character string, mail address to be used for replies.

Details

You need to have mailx installed and set up correctly for the mail delivery to work. Setup instructions on mailx are beyond the scope of this documentation.

Value

A character string, which can be fed to system() to actually send the mail.

Note

Try the heirloom-mailx package instead of bsd-mailx on Debian based distros.

Examples

```
exampleMail <- mailx("foo@example.com", "this is a test!",
  "hi,\n\ni'm just testing.\n\nbye\n", "me@example.com")
# inspect the generated commands
cat(exampleMail)
## Not run:
# evaluate commands, i.e., send the mail
system(exampleMail, intern=TRUE)

## End(Not run)
```

`qprint`*Quoted-printable encoding/decoding of text*

Description

This function is a simple wrapper for the `qprint[1]` command line tool to encode/decode messages in quoted-printable format. `qprint` must be installed for this to work.

Usage

```
qprint(text, encode = TRUE, qprint = Sys.which("qprint"), adjustLF = TRUE)
```

Arguments

<code>text</code>	Character string, the text to be encoded/decoded.
<code>encode</code>	Logical, if FALSE will run in decoding mode.
<code>qprint</code>	Character string, path to the <code>qprint</code> executable.
<code>adjustLF</code>	Logical, if TRUE, "\r" will be replaced with "\n" for convenience (e.g., see output by using <code>cat()</code>).

Value

Character string of the processed text.

`show,email-method`*Show method for S4 objects of class email*

Description

Used to display objects of class `email`.

Usage

```
## S4 method for signature 'email'
show(object)
```


Arguments

object An object of class email

See Also

[email](#)

Index

*Topic **classes**

email, -class, 2

*Topic **methods**

mail, 4

show, email-method, 8

*Topic **package**

mailR-package, 2

email, 4, 5, 8, 9

email, -class, 2

email-class (email, -class), 2

genMail, 3, 3, 5

mail, 3, 4, 4, 6

mail, -methods (mail), 4

mail, email-method (mail), 4

mailR-package, 2

mailx, 7

qprint, 4, 8

show, -methods (show, email-method), 8

show, email-method, 8