

- Content of this page was obsoleted.

x Internals: mail.pm

This module deals with mail sending and does the SMTP job. It provides a function for message distribution to a list, the message can be encrypted. There is also a function to send service messages by parsing tt2 files, These messages can be signed. For sending, a call to sendmail is done or the message is pushed in a spool according to calling context.

public functions

mail_file(), mail_message(), mail_forward(), set_send_spool(), reaper().

mail_file()

Message is written by parsing a tt2 file (or with a string). It writes mail headers if they are missing and they are encoded. Then the message is sent by calling mail::sending() function (see [mail-sending](#)).

IN:

1. filename: string - tt2 filename - no tt2 filename sent
2. rcpt (+): SCALAR ref(ARRAY) - SMTP RCPT To: field
3. data (+): ref(HASH) - used to parse tt2 file, contains header values, keys are:
 - return_path (+): SMTP MAIL From: field *if send by SMTP*, q X-Sympa-From: field *if send by spool*
 - to: To: header field *else it is \$rcpt*
 - from: From: field *if \$filename is not a full msg*
 - subject: Subject: field *if \$filename is not a full msg*
 - replyto: Reply-to: field *if \$filename is not a full msg*
 - headers: ref(HASH), keys are other mail headers
 - body: body message *if not \$filename*
 - lang: tt2 language *if \$filename*
 - list: ref(HASH) *if sign_mode='smime'* - keys are:
 - name: list name
 - dir: list directory
4. robot (+): robot
5. sign_mode: 'smime'- the mail is signed with smime undef - no signature

OUT: 1

mail_message()

Distributes a message to a list. The message is encrypted if needed, in this case, only one SMTP

session is used for each recipient otherwise, recipients are grouped by domain for sending (it controls the number of recipient arguments to call `sendmail`). Message is sent by calling `mail::sendto()` function (see [mail-sendto](#)).

IN:

1. `message (+)`: `ref(Message)` - message to distribute
2. `from (+)`: message from
3. `robot (+)`: robot
4. `rcpt (+)`: ARRAY - recipients

OUT: `$numsmtp` = number of `sendmail` processes used

mail_forward()

Forward a message by calling `mail::sending()` function (see [mail-sending](#)).

IN:

1. `msg (+)`: `ref(Message)` `ref(MIME::Entity)` string - message to forward
2. `from (+)`: message from
3. `rcpt (+)`: `ref(SCALAR)` `ref(ARRAY)` - recipients
4. `robot (+)`: robot

OUT: 1 undef

set_send_spool()

Used by other processes than `sympa.pl` to indicate to send message by writing message in spool instead of calling `mail::smtpd()` function (see [mail-smtpd](#)). The concerned spool is set in `$send_spool` global variable, used by `mail::sending()` function (see [mail-sending](#)).

IN:

1. `spool (+)`: the concerned spool for sending.

OUT: -

reaper()

Non blocking function used to clean the defunct child processes by waiting for them and then decreasing the counter. For example, this function is called by `mail::smtpd()` (see [mail-smtpd](#)), main loop of `sympa.pl`, `task_manager.pl`, `bounced.pl`.

IN:

1. block

OUT: the pid of the defunct process -1 if there is no such child process.

private functions

sendto(), sending(), smtpo().

sendto()

Encodes subject header. Encrypts the message if needed. In this case, it checks if there is only one receipient. Then the message is sent by calling mail::sending() function (see [mail-sending](#)).

IN:

1. msg_header (+): ref(MIME::Head) - message header
2. msg_body (+): message body
3. from (+): message from
4. rcpt (+): ref(SCALAR) ref(ARRAY) - message receipients (ref(SCALAR) for encryption)
5. robot (+): robot
6. encrypt: 'smime_crypted' the mail is encrypted with smime undef - no encryption

OUT: 1 - sending by calling smtpo (sendmail) 0 - sending by push in spool undef

sending()

Signs the message according to \$sign_mode. Chooses sending mode according to context. If \$send_spool global variable is empty, the message is sent by calling mail::smtpo() function (see [mail-smtpo](#)) else the message is written in spool \$send_spool in order to be handled by sympa.pl process (because only this is allowed to make a fork). When the message is pushed in spool, these mail headers are added:

- X-Sympa-To:: receipients
- X-Sympa-From:: from
- X-Sympa-Checksum:: to check allowed program to push in spool

A message pushed in spool like this will be handled later by sympa::DoSendMessage() function (see [sympa-dosendmessage](#))

IN:

1. msg (+): ref(MIME::Entity) string - message to send
2. rcpt (+): ref(SCALAR) ref(ARRAY) - receipients (for SMTP: RCPT To: field)
3. from (+): for SMTP: MAIL From: field for spool sending: X-Sympa-From field
4. robot (+): robot
5. listname: listname "
6. sign_mode (+): 'smime' 'none'for signing
7. sympa_email: for the file name for spool sending

OUT: 1 - sending by calling smtpo() (sendmail) 0 - sending by pushing in spool undef

smtpto()

Calls to sendmail for the recipients given as argument by making a fork and an exec. Before, waits for number of children process number allowed by sympa.conf by calling mail::reaper() function (see [mail-reaper](#)).

IN:

1. from (+): SMTP MAIL From: field
2. rcpt (+): ref(SCALAR)) ref(ARRAY) - SMTP RCPT To: field
3. robot (+): robot

OUT: mail:: \$fh - file handle on opened file for ouput, for SMTP DATA field undef

From:
<https://www.sympa.org/> - **Sympa mailing list server**

Permanent link:
<https://www.sympa.org/internals/internals-mail>

Last update: **2018/03/27 10:06**

