gs.email Documentation

Release 2.2.0

GroupServer.org

April 28, 2016

Contents

This is the core product for *sending* email from GroupServer via SMTP [1]. It is used by the groups to send email to the group members [2], and the user-profile system to send notifications [3].

Contents:

[1] *Receiving* email is supported by the `gs.group.messages.add.base` product <https://github.com/groupserver/gs.group.messages.add.base>
[2] *Sending* email from groups is handled by the `gs.group.list.sender` product <https://github.com/groupserver/gs.group.list.sender>
[3] Notifications are sent by the `gs.profile.notify` product <https://github.com/groupserver/gs.profile.notify/>

CHAPTER 1

Configuration

The configuration for sending email is controlled by the `gsconfig.ini` file. The configuration *options* set up how the system connects to the SMTP server [1]. Delivery of email messages can be can be to a local server, a remote server, or turned off entirely as shown in the *configuration examples* below.

1.1 Options

`hostname` (required): The name of the SMTP server (`localhost` if the SMTP server is running on the same machine as GroupServer).

`port` (required): The port that the SMTP server runs on (usually `25`).

`username` (optional): The name of the user that logs into the SMTP server to send the message. (Defaults to `None`.)

`password` (optional): The password used to log into the SMTP server. (Defaults to `None`.)

`no_tls` and `force_tls` (both optional): Transport Layer Security (TLS) is the replacement to the Secure Sockets Layer (SSL). It can be used to encrypt the communication between GroupServer and the SMTP server. Normally the system will use TLS if it is available.

Setting the `no_tls` option to `False` will force the GroupServer to connect to the SMTP server *en clear*, even if encryption is available. This may be useful if the SMTP server only accepts connections from `localhost` and it is running on the same machine as GroupServer.

Setting the `force_tls` to `True` forces GroupServer to use encryption to connect to the SMTP server. If TLS is not available then a `RuntimeError` is raised.

`queuepath` (optional): The path to the `Maildir` folder that stores all the messages before processing by the SMTP server. Defaults to `/tmp/mailqueue`.

`processorthread` (optional): If `True` (the default) then a separate thread will be started to handle the queue and pass the email messages on to the SMTP server. If `False` the email messages will just be written to the file in `queuepath` and not be processed (which is **very** useful for testing).

`xverp` (optional): If `True` then XVERP will be used when the email messages are sent [2].

1.2 Examples

Setting up delivery to the local SMTP server, from the GroupServer instance called `main`:

[1] Configuration is handled by the `gs.email.config` module. It uses the `gs.config` module to read the configuration information <https://github.com/groupserver/gs.config>

[2] For more information about XVERP see The Postfix VERP Howto.

```
[config-main]
smtp = local

[smtp-local]
hostname = localhost
port = 25
no_tls = True
queuepath = /tmp/main-mail-queue
xverp = True
```

Note There will be more than the `smtp` option for the configuration of the `main` GroupServer instance. However, the other options have been left out for clarity.

Setting up delivery to a remote SMTP server, from the GroupServer instance called `production`:

```
[config-production]
smtp = remote

[smtp-remote]
hostname = remote.host.name
port = 2525
username = user_on_the_remote_server
password = password_on_the_remote_server
force_tls = True
queuepath = /tmp/production-mail-queue
processorthread = True
xverp = True
```

Setting up a test system to not send out email:

```
[config-test]
smtp = none

[smtp-none]
hostname = localhost
port = 25
queuepath = /tmp/test-mail-queue
processorthread = False
```

CHAPTER 2

Troubleshooting

If mail is trapped in `queuedir/new` look to see if `.sending_*` or `.rejected_*` files have been created in the same directory. If so, delete them and the mail should be processed.

CHAPTER 3

gs.email API Reference

The main function used by external code in the *send_email()* function. Internally it uses the *mailer* to send messages based on the *configuration*.

3.1 send_email

gs.email.**send_email**(*sender*, *recipients*, *email*)
Send an email message to some recipients

Parameters

- **sender** (*str*) – The address of the person, or group, that is responsible for sending the email message. This will become the from-address on the *envelope;* it is separate from the *From*, *Sender*, and *Reply-to* addresses in the email message.
- **recipients** (*str, tuple, list*) – The address of the person who should receive the email message, a list of recipients, or a tuple containing the addresses of the recipients. This will become the to-address on the *envelope;* it is separate from the To, CC, and BCC addresses in the email message.
- **email** (*str*) – The email message, as a string. It needs to be a complete message with headers and a body.

Returns None.

The *send_email()* function uses SMTP to send an email message to the recipients, from the sender, in *batches* of *gs.email.core.MAX_BATCH* recipients. The batching is necessary to prevent overwhelming the SMTP server (it makes management of the mail queue easier).

gs.email.core.**MAX_BATCH = 50**
The maximum number of email recipients in a batch.

3.1.1 Examples

Send an email from the support-address of the site to all the addresses of a GroupServer user:

```
eu = gs.profile.email.base.EmailUser(context, userInfo)
send_email(siteInfo.get_support_email(), eu.get_addresses(), emailMessage)
```

The gs.profile.notify.NotifyUser class demonstrates how to send an email message using *send_email()*. The gs.profile.notify.MessageSender class demonstrates how an email message is constructed using the standard Python email module.

3.2 Mailer

The *gs.email.mailer.XVERPSMTPMailer* is loaded when the configuration option `xverp` is set to `True` (see Configuration). As its name implies, it turns on XVERP, so the groups can be informed when an address bounces [1]. For the most part the mailer is the same as that provided by zope.sendmail.

class `gs.email.mailer.`**`XVERPSMTPMailer`**(*hostname='localhost'*, *port=25*, *username=None*, *password=None*, *no_tls=False*, *force_tls=False*)

Sending messages to an SMTP server using TLS and XVERP

`send`(*fromaddr*, *toaddrs*, *message*)

Send a message

Parameters

- **`fromaddr`** (*str*) – The envelope-from.
- **`toaddrs`** (*list*) – The envelope-to addresses.
- **`message`** (*str*) – The email message to send.

Returns `None`

`send()` will send a message to the SMTP server, requesting that XVERP is used. This is effectively the same as the `zope.sendmail.mailer.SMTPMailer.send()` method, except `mail_options` is used to pass `XVERP` to the SMTP server. TLS is used where possible.

3.3 Configuration

The *gs.email.config.create_emailUtilities()* function loads the configuration used to connect to the outgoing SMTP server, before loading an appropriate *mailer*.

`gs.email.config.`**`create_emailUtilities`**(*instance_id=None*)

Create the utilities to send the email messages

Parameters **`instance_id`** (*str*) – The indentifier for the GroupServer instance

Returns `None`

The `create_emailUtilities()` function loads the `smtp` section of the configuration of the instance specified by `instance_id`. If no instance is specified then `gs.config.getInstanceId()` is used to determine the current instance. It then loads the following configuration options:

- `hostname`
- `port`
- `username`
- `password`
- `no_tls`
- `force_tls`
- `queuepath`
- `processorthread`
- `xverp`

[1] For more information about XVERP see *The Postfix VERP Howto* <http://www.postfix.org/VERP_README.html>

If the XVERP option is `True` then *gs.email.mailer.XVERPSMTPMailer* is registered as the utility used to connect to the SMTP host; otherwise `zope.sendmail.mailer.SMTPMailer` is used. In either case the mailer is configured with the options in the config file.

CHAPTER 4

Changelog

4.1 2.2.0 (2015-03-17)

- Turning on lax-parsing of the config, to avoid issues with the presence (or absence) of the relay-address-prefix

4.2 2.1.2 (2014-10-24)

- Using GitHub as the canonical repository
- Naming the reStructuredText files as such

4.3 2.1.1 (2014-03-21)

- Adding a try-except block

4.4 2.1.0 (2014-01-24)

- Cleaning up the imports
- Cleaning up the code to make it **PEP 8** compliant

4.5 2.0.1 (2012-07-28)

- Fixing a typing mistake

4.6 2.0.0 (2012-07-19)

Initial version. Prior to this sending email was carried out by some code in the ZMI.

CHAPTER 5

Resources

- Code repository: https://github.com/groupserver/gs.email/
- Documentation: http://groupserver.readthedocs.io/projects/gsemail/
- Questions and comments to http://groupserver.org/groups/development/
- Report bugs at https://redmine.iopen.net/projects/groupserver/

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

C

M

P

S

X