
gs.group.list.base Documentation

Release 1.1.1

GroupServer.org

April 28, 2016

1	Email message	3
2	HTML to Text	7
2.1	Example	7
3	Reply to	9
4	Changelog	11
4.1	1.1.1 (2015-12-10)	11
4.2	1.1.0 (2015-09-24)	11
4.3	1.0.3 (2015-04-08)	11
4.4	1.0.2 (2015-02-11)	11
4.5	1.0.1 (2015-02-10)	11
4.6	1.0.0 (2015-01-23)	11
5	Indices and tables	13

This product supplies the code common to all the list-products. A surprisingly large amount of [GroupServer](#) functionality has absolutely nothing to do with what people normally consider core mailing-list functionality. However, the `gs.group.list.*` products do.

Contents:

Email message

The `EmailMessage` class represents an email message that is being added to a group. It takes an email message, with its lies about character encodings, and produces a message with a Unicode plain-text body, a Unicode HTML-formatted body (if present) and some headers in Unicode. (Most of the attributes of the `EmailMessage` are methods decorated with the `zope.cachedescriptors.property.Lazy()` decorator.)

```
class gs.group.list.base.EmailMessage(messageString, list_title='', group_id='', site_id='',
                                     sender_id_cb=None)
```

An email message with Unicode knowledge

Parameters

- **messageString** (*str*) – The email message.
- **listTitle** (*str*) – The name of the group.
- **group_id** (*str*) – The identifier for the group.
- **site_id** (*str*) – The identifier for the site that contains the group.
- **sender_id_cb** (*function*) – The function to call to get the identifier of the message author from an email address.

The standard Python `email.message.Message` class is great. Really. Use it. About the only thing it lacks is some nouse about GroupServer groups, and it does not provide Unicode versions of the headers by default.

message

Return type `email.message.Message`

The parsed version of the `messageString`.

encoding

Return type `unicode`

The encoding of the message, or `utf-8` if the encoding is lies.

attachments

The files attached to the message, **including** the HTML and plain-text bodies, but **excluding** those that lack filenames.

The attachments are represented as dictionaries with the following values.

payload:

The content of the attachment, decoded (based on the `Content-Transfer-Encoding`) to a sequence of bytes.

fileid:

The GroupServer file identifier.

filename:

The name of the file. [RFC 2183#2.3](#) restricts this to ASCII, but the value is Unicode.

length:

The length of the file in bytes.

charset

The character-set of the file if the major-type of the MIME-type of the attachment is `text`; `None` for all other MIME types. If the attachment itself does not specify a character-set then the character-set for the overall message is returned. If the overall message does not specify a character-set then `utf-8` is returned.

mimetype:

The MIME type of the file.

maintype:

The main-type of the `mimetype`.

subtype:

The subtype of the `mimetype`.

contentid:

The value of the `Content-ID` header for the attachment, or an empty string if absent.

body

Return type `unicode`

The plain-text (`text/plain`) version of the message body, decoded into a `unicode` string. If absent (which happens sometimes) the `EmailMessage.html_body` is converted to plain text and returned.

html_body

Return type `unicode`

The HTML version (`text/html`) of the message body, decoded into a `unicode` string. If absent an empty string (`' '`) is returned.

subject

Return type `unicode`

The *Subject* of the message, without the group name, and `Re :`.

It is common for a *Subject* of an email to contain the name of the group that the message is from, or is posted-to:

Subject: [groupserver development] Email Processing Rewrite

While useful for the recipients of the message, it is just noise for GroupServer.

compressed_subject

Return type `unicode`

The `EmailMessage.subject` without white-space, and transformed to lowercase, which is useful for comparisons (see `EmailMessage.topic_id`).

sender

Return type `unicode`

The email address of the person who wrote the message. This is actually generated from the *From* header, rather than the *Sender* header.

name**Return type** `unicode`

The name of the person who wrote the message, taken from the *From* header.

topic_id**Return type** `unicode`

The identifier of the topic that this post will belong to.

A *topic_id* for two posts will clash if the *EmailMessage.compressed_subject*, group identifier, and site identifier are all identical.

post_id**Return type** `unicode`

The identifier for the post, which will (almost certainly) be unique.

A *post_id* for a post will clash with another post if

- The *topic_id* is the same, so
 - The *compressed_subject* is the same, and
 - The group identifier is the same, and
 - The site identifier is the same, and
- The *body* of the post is the same, and
- The *sender* is the same author, and
- The post is a response to the same message (the value of the *In-Reply-To* header is the same), and
- The total length of all the attachments is the same.

HTML to Text

The `gs.group.list.base.html2txt.HTMLConverter` class is a subclass of `HTMLParser.HTMLParser` (or `html.parser.HTMLParser` in Python 3) that produces a plain-text version of a HTML documents. It is fairly simple, returning a Unicode version of the HTML, and it is used in the rare case that a plain-text body is absent from an email message.

The `convert_to_txt()` function is a wrapper for convenience.

2.1 Example

```
>>> from gs.group.list.base.html2txt import HTMLConverter
>>> converter = HTMLConverter()
>>> html = '<p>Je ne ecrit pas fran&ccedil;ais.</p>'
>>> converter.feed(html)
>>> converter.close()
>>> print(converter)
Je ne ecrit pas français.
```

Reply to

The `ReplyTo` enumeration lists the different settings that a *Reply-to* header can have, while the `replyto()` function returns the current setting for a mailing list.

Changelog

4.1 1.1.1 (2015-12-10)

- Fixing the unit-tests so the work with `zc.recipe.testrunner`

4.2 1.1.0 (2015-09-24)

- Adding the `replyto` function, and `ReplyTo` enumeration, moving them here from `gs.group.list.sender`
- Updating the documentation

4.3 1.0.3 (2015-04-08)

- Improving the handling of `Subject` headers that contain `[square brackets]`

4.4 1.0.2 (2015-02-11)

- Handling corner cases where the header lies about its encoding

4.5 1.0.1 (2015-02-10)

- Handling corner cases where the body or HTML body is `None`

4.6 1.0.0 (2015-01-23)

Initial release. Prior to the creation of `this product` the code was found in the `Products.XWFMailinListManager` product.

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

-
- Documentation: <http://groupserver.readthedocs.io/projects/gsgrouplistbase/>
 - Code repository: <https://github.com/groupserver/gs.group.list.base>
 - Questions and comments to <http://groupserver.org/groups/development>
 - Report bugs at <https://redmine.iopen.net/projects/groupserver>

A

attachments (gs.group.list.base.EmailMessage attribute), 3

B

body (gs.group.list.base.EmailMessage attribute), 4

C

compressed_subject (gs.group.list.base.EmailMessage attribute), 4

E

EmailMessage (class in gs.group.list.base), 3

encoding (gs.group.list.base.EmailMessage attribute), 3

H

html_body (gs.group.list.base.EmailMessage attribute), 4

M

message (gs.group.list.base.EmailMessage attribute), 3

N

name (gs.group.list.base.EmailMessage attribute), 5

P

post_id (gs.group.list.base.EmailMessage attribute), 5

R

RFC

RFC 2183#2.3, 4

S

sender (gs.group.list.base.EmailMessage attribute), 4

subject (gs.group.list.base.EmailMessage attribute), 4

T

topic_id (gs.group.list.base.EmailMessage attribute), 5