`!email.encoders`: Encoders

email.encoders

Source code: `Lib/email/encoders.py`

This module is part of the legacy (`Compat32`) email API. In the new API the functionality is provided by the *cte* parameter of the `~email.message.EmailMessage.set_content` method.

This module is deprecated in Python 3. The functions provided here should not be called explicitly since the `~email.mime.text.MIMEText` class sets the content type and CTE header using the *_subtype* and *_charset* values passed during the instantiation of that class.

The remaining text in this section is the original documentation of the module.

When creating `~email.message.Message` objects from scratch, you often need to encode the payloads for transport through compliant mail servers. This is especially true for `image/\*` and `text/\*` type messages containing binary data.

The `email` package provides some convenient encoders in its `!encoders` module. These encoders are actually used by the `~email.mime.audio.MIMEAudio` and `~email.mime.image.MIMEImage` class constructors to provide default encodings. All encoder functions take exactly one argument, the message object to encode. They usually extract the payload, encode it, and reset the payload to this newly encoded value. They should also set the `Content-Transfer-Encoding` header as appropriate.

Note that these functions are not meaningful for a multipart message. They must be applied to individual subparts instead, and will raise a `TypeError` if passed a message whose type is multipart.

Here are the encoding functions provided:

encode_quopri(msg)

Encodes the payload into quoted-printable form and sets the `Content-Transfer-Encoding` header to `quoted-printable`[^1]. This is a good encoding to use when most of your payload is normal printable data, but contains a few unprintable characters.

encode_base64(msg)

Encodes the payload into base64 form and sets the `Content-Transfer-Encoding` header to `base64`. This is a good encoding to use when most of your payload is unprintable data since it is a more compact form than quoted-printable. The drawback of base64 encoding is that it renders the text

non-human readable.

encode_7or8bit(msg)

This doesn't actually modify the message's payload, but it does set the `Content-Transfer-Encoding` header to either `7bit` or `8bit` as appropriate, based on the payload data.

encode_noop(msg)

This does nothing; it doesn't even set the `Content-Transfer-Encoding` header.

Footnotes

[^1]: Note that encoding with `encode_quopri` also encodes all tabs and space characters in the data.

□□□ #2
□ □□□ □□□ 2026-10-31 17:08:56 UTC
□ □□□ □□□ 2026-10-31 17:08:57 UTC