:mod:`email.utils`: Miscellaneous utilities

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 1); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 4)

Unknown directive type "module".

```
.. module:: email.utils
   :synopsis: Miscellaneous email package utilities.
```

Source code: :source:`Lib/email/utils.py`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 7); *backlink*

Unknown interpreted text role "source".

There are a couple of useful utilities provided in the :mod:`email.utils` module:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 11); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 14)

Unknown directive type "function".

```
.. function:: localtime(dt=None)

    Return local time as an aware datetime object.  If called without
    arguments, return current time.  Otherwise *dt* argument should be a
    :class:`~datetime.datetime` instance, and it is converted to the local time
    zone according to the system time zone database.  If *dt* is naive (that
    is, ``dt.tzinfo`` is ``None``), it is assumed to be in local time.  In this
    case, a positive or zero value for *isdst* causes ``localtime`` to presume
    initially that summer time (for example, Daylight Saving Time) is or is not
    (respectively) in effect for the specified time.  A negative value for
    *isdst* causes the ``localtime`` to attempt to divine whether summer time
    is in effect for the specified time.

    .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 30)

Unknown directive type "function".

```
.. function:: make_msgid(idstring=None, domain=None)

   Returns a string suitable for an :rfc:`2822`\ -compliant
   :mailheader:`Message-ID` header.  Optional *idstring* if given, is a string
   used to strengthen the uniqueness of the message id.  Optional *domain* if
   given provides the portion of the msgid after the '@'.  The default is the
   local hostname.  It is not normally necessary to override this default, but
   may be useful certain cases, such as a constructing distributed system that
   uses a consistent domain name across multiple hosts.

   .. versionchanged:: 3.2
      Added the *domain* keyword.
```

The remaining functions are part of the legacy (`Compat32`) email API. There is no need to directly use these with the new API, since the parsing and formatting they provide is done automatically by the header parsing machinery of the new API.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 50)

Unknown directive type "function".

```
.. function:: quote(str)

   Return a new string with backslashes in *str* replaced by two backslashes, and
   double quotes replaced by backslash-double quote.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 56)

Unknown directive type "function".

```
.. function:: unquote(str)

   Return a new string which is an *unquoted* version of *str*. If *str* ends and
   begins with double quotes, they are stripped off.  Likewise if *str* ends and
   begins with angle brackets, they are stripped off.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 63)

Unknown directive type "function".

```
.. function:: parseaddr(address)

   Parse address -- which should be the value of some address-containing field such
   as :mailheader:`To` or :mailheader:`Cc` -- into its constituent *realname* and
   *email address* parts.  Returns a tuple of that information, unless the parse
   fails, in which case a 2-tuple of ``('', '')`` is returned.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 71)

Unknown directive type "function".

```
.. function:: formataddr(pair, charset='utf-8')

   The inverse of :meth:`parseaddr`, this takes a 2-tuple of the form ``(realname,
   email_address)`` and returns the string value suitable for a :mailheader:`To` or
   :mailheader:`Cc` header.  If the first element of *pair* is false, then the
   second element is returned unmodified.

   Optional *charset* is the character set that will be used in the :rfc:`2047`
   encoding of the ``realname`` if the ``realname`` contains non-ASCII
   characters.  Can be an instance of :class:`str` or a
   :class:`~email.charset.Charset`.  Defaults to ``utf-8``.

   .. versionchanged:: 3.3
      Added the *charset* option.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 87)

Unknown directive type "function".

```
.. function:: getaddresses(fieldvalues)

   This method returns a list of 2-tuples of the form returned by ``parseaddr()``.
   *fieldvalues* is a sequence of header field values as might be returned by
   :meth:`Message.get_all <email.message.Message.get_all>`.  Here's a simple
```

example that gets all the recipients of a message::

```
from email.utils import getaddresses

tos = msg.get_all('to', [])
ccs = msg.get_all('cc', [])
resent_tos = msg.get_all('resent-to', [])
resent_ccs = msg.get_all('resent-cc', [])
all_recipients = getaddresses(tos + ccs + resent_tos + resent_ccs)
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 103)

Unknown directive type "function".

```
.. function:: parsedate(date)

   Attempts to parse a date according to the rules in :rfc:`2822`. however, some
   mailers don't follow that format as specified, so :func:`parsedate` tries to
   guess correctly in such cases.  *date* is a string containing an :rfc:`2822`
   date, such as  ``"Mon, 20 Nov 1995 19:12:08 -0500"``.  If it succeeds in parsing
   the date, :func:`parsedate` returns a 9-tuple that can be passed directly to
   :func:`time.mktime`; otherwise ``None`` will be returned.  Note that indexes 6,
   7, and 8 of the result tuple are not usable.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 114)

Unknown directive type "function".

```
.. function:: parsedate_tz(date)

   Performs the same function as :func:`parsedate`, but returns either ``None`` or
   a 10-tuple; the first 9 elements make up a tuple that can be passed directly to
   :func:`time.mktime`, and the tenth is the offset of the date's timezone from UTC
   (which is the official term for Greenwich Mean Time) [#]_.  If the input string
   has no timezone, the last element of the tuple returned is ``0``, which represents
   UTC. Note that indexes 6, 7, and 8 of the result tuple are not usable.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 124)

Unknown directive type "function".

```
.. function:: parsedate_to_datetime(date)

   The inverse of :func:`format_datetime`.  Performs the same function as
   :func:`parsedate`, but on success returns a :mod:`~datetime.datetime`;
   otherwise ``ValueError`` is raised if *date* contains an invalid value such
   as an hour greater than 23 or a timezone offset not between -24 and 24 hours.
   If the input date has a timezone of ``-0000``, the ``datetime`` will be a naive
   ``datetime``, and if the date is conforming to the RFCs it will represent a
   time in UTC but with no indication of the actual source timezone of the
   message the date comes from.  If the input date has any other valid timezone
   offset, the ``datetime`` will be an aware ``datetime`` with the
   corresponding a :class:`~datetime.timezone` :class:`~datetime.tzinfo`.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 140)

Unknown directive type "function".

```
.. function:: mktime_tz(tuple)

   Turn a 10-tuple as returned by :func:`parsedate_tz` into a UTC
   timestamp (seconds since the Epoch).  If the timezone item in the
   tuple is ``None``, assume local time.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 147)

Unknown directive type "function".

```
.. function:: formatdate(timeval=None, localtime=False, usegmt=False)

   Returns a date string as per :rfc:`2822`, e.g.::

      Fri, 09 Nov 2001 01:08:47 -0000

   Optional *timeval* if given is a floating point time value as accepted by
   :func:`time.gmtime` and :func:`time.localtime`, otherwise the current time is
   used.

   Optional *localtime* is a flag that when ``True``, interprets *timeval*, and
   returns a date relative to the local timezone instead of UTC, properly taking
   daylight savings time into account. The default is ``False`` meaning UTC is
   used.

   Optional *usegmt* is a flag that when ``True``, outputs a  date string with the
   timezone as an ascii string ``GMT``, rather than a numeric ``-0000``. This is
   needed for some protocols (such as HTTP). This only applies when *localtime* is
   ``False``.  The default is ``False``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 168)

Unknown directive type "function".

```
.. function:: format_datetime(dt, usegmt=False)

   Like ``formatdate``, but the input is a :mod:`datetime` instance.  If it is
   a naive datetime, it is assumed to be "UTC with no information about the
   source timezone", and the conventional ``-0000`` is used for the timezone.
   If it is an aware ``datetime``, then the numeric timezone offset is used.
   If it is an aware timezone with offset zero, then *usegmt* may be set to
   ``True``, in which case the string ``GMT`` is used instead of the numeric
   timezone offset.  This provides a way to generate standards conformant HTTP
   date headers.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 182)

Unknown directive type "function".

```
.. function:: decode_rfc2231(s)

   Decode the string *s* according to :rfc:`2231`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 187)

Unknown directive type "function".

```
.. function:: encode_rfc2231(s, charset=None, language=None)

   Encode the string *s* according to :rfc:`2231`.  Optional *charset* and
   *language*, if given is the character set name and language name to use.  If
   neither is given, *s* is returned as-is.  If *charset* is given but *language*
   is not, the string is encoded using the empty string for *language*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-

main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 195)

Unknown directive type "function".

```
.. function:: collapse_rfc2231_value(value, errors='replace', fallback_charset='us-ascii')

   When a header parameter is encoded in :rfc:`2231` format,
   :meth:`Message.get_param <email.message.Message.get_param>` may return a
   3-tuple containing the character set,
   language, and value.  :func:`collapse_rfc2231_value` turns this into a unicode
   string.  Optional *errors* is passed to the *errors* argument of :class:`str`'s
   :func:`~str.encode` method; it defaults to ``'replace'``.  Optional
   *fallback_charset* specifies the character set to use if the one in the
   :rfc:`2231` header is not known by Python; it defaults to ``'us-ascii'``.

   For convenience, if the *value* passed to :func:`collapse_rfc2231_value` is not
   a tuple, it should be a string and it is returned unquoted.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.utils.rst, line 210)

Unknown directive type "function".

```
.. function:: decode_params(params)

   Decode parameters list according to :rfc:`2231`.  *params* is a sequence of
   2-tuples containing elements of the form ``(content-type, string-value)``.
```

Footnotes

[1] Note that the sign of the timezone offset is the opposite of the sign of the `time.timezone` variable for the same timezone; the latter variable follows the POSIX standard while this module follows RFC 2822.