NAME

Mail::Message::Convert::Html − Format messages in HTML

INHERITANCE

```
 Mail::Message::Convert::Html
   is a Mail::Message::Convert
   is a Mail::Reporter
```

SYNOPSIS

```
 use Mail::Message::Convert::Html;
 my $Html = Mail::Message::Convert::Html−>new;

 print $html−>fieldToHtml($head);
 print $html−>headToHtmlHead($head);
 print $html−>headToHtmlTable($head);
 print $html−>textToHtml($text);
```

DESCRIPTION

The package contains various translators which handle HTML or XHTML without the help of external modules. There are more HTML related modules, which do require extra packages to be installed.

Extends "DESCRIPTION" in Mail::Message::Convert.

METHODS

Extends "METHODS" in Mail::Message::Convert.

Constructors

Extends "Constructors" in Mail::Message::Convert.

Mail::Message::Convert::Html−>**new**(%options)

```
 −Option     −−Defined in             −−Default
  fields       Mail::Message::Convert  <see description>
  head_mailto                          <true>
  log          Mail::Reporter          'WARNINGS'
  produce                              HTML
  trace        Mail::Reporter          'WARNINGS'
```

fields => NAMES|ARRAY−OF−NAMES|REGEXS

head_mailto => BOOLEAN

Whether to replace e−mail addresses in some header lines with links.

log => LEVEL

produce => 'HTML'|'XHTML'

Produce HTML or XHTML output. The output is slightly different, even html browsers will usually accept the XHTML data.

trace => LEVEL

Converting

Extends "Converting" in Mail::Message::Convert.

`$obj`−>**fieldContentsToHtml**(`$field`, [$subject])

Format one field from the header to HTML. When the header line usually contains e−mail addresses, the line is scanned and valid addresses are linked with an `mailto:` anchor. The`$subject` can be specified to be included in that link.

`$obj`−>**fieldToHtml**(`$field`, [$subject])

Reformat one header line field to HTML. The`$field`' s name is printed in bold, followed by the formatted field content, which is produced by **fieldContentsToHtml()**.

`$obj`−>**headToHtmlHead**($head, `$meta`)

Translate the selected header lines (fields) to an html page header. Each selected field will get its own meta line with the same name as the line. Furthermore, the `Subject` field will become the `title`, and `From` is used for the `Author`.

Besides, you can specify your own meta fields, which will overrule header fields. Empty fields will not be included. When a `title` is specified, this will become the html title, otherwise the `Subject` field is taken. In list context, the lines are separately, where in scalar context the whole text is returned as one.

If you need to add lines to the head (for instance, http-equiv lines), then splice them before the last element in the returned list.

example:

```
my @head = $html->headToHtmlHead
    ( $head
    , description => 'This is a message'
    , generator   => 'Mail::Box'
    );
splice @head, -1, 0, '<meta http-equiv=...>';
print @head;
```

`$obj`->**headToHtmlTable**(`$head`, [$table_params])
Produce a display of the **selectedFields()** of the header in a table shape. The optional `$table_params` are added as parameters to the produced TABLE tag. In list context, the separate lines are returned. In scalar context, everything is returned as one.

example:

```
print $html->headToHtmlTable($head, 'width="50%"');
```

`$obj`->**selectedFields**($head)
Inherited, see "Converting" in Mail::Message::Convert

`$obj`->**textToHtml**($lines)
Translate one or more `$lines` from text into HTML. Each line is taken one after the other, and only simple things are translated. `textToHtml` is able to con vert large plain texts in a descent fashion. In scalar context, the resulting lines are returned as one.

Error handling

Extends "Error handling" in Mail::Message::Convert.

`$obj`->**AUTOLOAD**()
Inherited, see "Error handling" in Mail::Reporter

`$obj`->**addReport**($object)
Inherited, see "Error handling" in Mail::Reporter

`$obj`->**defaultTrace**([$level]|[$loglevel, `$tracelevel`]|[$level, `$callback`])
Mail::Message::Convert::Html->**defaultTrace**([$level]|[$loglevel, `$tracelevel`]|[$level, `$callback`])
Inherited, see "Error handling" in Mail::Reporter

`$obj`->**errors**()
Inherited, see "Error handling" in Mail::Reporter

`$obj`->**log**([$level, [$strings]])
Mail::Message::Convert::Html->**log**([$level, [$strings]])
Inherited, see "Error handling" in Mail::Reporter

`$obj`->**logPriority**($level)
Mail::Message::Convert::Html->**logPriority**($level)
Inherited, see "Error handling" in Mail::Reporter

`$obj->`**logSettings**()
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**notImplemented**()
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**report**([$level])
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**reportAll**([$level])
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**trace**([$level])
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**warnings**()
Inherited, see ‘‘Error handling’’ in Mail::Reporter

Cleanup

Extends ‘‘Cleanup’’ in Mail::Message::Convert.

`$obj->`**DESTROY**()
Inherited, see ‘‘Cleanup’’ in Mail::Reporter

DIAGNOSTICS

Error: Package `$package` does not implement `$method`.
Fatal error: the specific package (or one of its superclasses) does not implement this method where it should. This message means that some other related classes do implement this method however the class at hand does not. Probably you should investigate this and probably inform the author of the package.

SEE ALSO

This module is part of Mail-Message distribution version 3.012, built on February 11, 2022. Website: *http://perl.overmeer.net/CPAN/*

LICENSE