

NAME

Mail::Message::Field::URIs – message header field with uris

INHERITANCE

```
Mail::Message::Field::URIs
  is a Mail::Message::Field::Structured
  is a Mail::Message::Field::Full
  is a Mail::Message::Field
  is a Mail::Reporter
```

SYNOPSIS

```
my $f = Mail::Message::Field->new('List-Post' => 'http://x.org/');

my $g = Mail::Message::Field->new('List-Post');
$g->addURI('http://x.org');

my $uri = URI->new(...);
$g->addURI($uri);

my @uris = $g->URIs;
```

DESCRIPTION

More recent RFCs prefer uri field notation over the various differentiated syntaxes. Especially the mailing-list RFCs use these fields all the time. This class can maintain these fields.

Extends “DESCRIPTION” in Mail::Message::Field::Structured.

OVERLOADED

Extends “OVERLOADED” in Mail::Message::Field::Structured.

overload: “”

Inherited, see “OVERLOADED” in Mail::Message::Field

overload: 0+

Inherited, see “OVERLOADED” in Mail::Message::Field

overload: <=>

Inherited, see “OVERLOADED” in Mail::Message::Field

overload: bool

Inherited, see “OVERLOADED” in Mail::Message::Field

overload: cmp

Inherited, see “OVERLOADED” in Mail::Message::Field

overload: stringification

Inherited, see “OVERLOADED” in Mail::Message::Field::Full

METHODS

Extends “METHODS” in Mail::Message::Field::Structured.

Constructors

Extends “Constructors” in Mail::Message::Field::Structured.

`$obj->clone()`

Inherited, see “Constructors” in Mail::Message::Field

`Mail::Message::Field::URIs->from($field, %options)`

Inherited, see “Constructors” in Mail::Message::Field::Full

`Mail::Message::Field::URIs->new($data)`

--Option	--Defined in	--Default
attributes	Mail::Message::Field::Structured	<ignored>
charset	Mail::Message::Field::Full	undef
datum	Mail::Message::Field::Structured	undef
encoding	Mail::Message::Field::Full	'q'
force	Mail::Message::Field::Full	false
language	Mail::Message::Field::Full	undef
log	Mail::Reporter	'WARNINGS'
trace	Mail::Reporter	'WARNINGS'

attributes => ATTRS

charset => STRING

datum => STRING

encoding => 'q'|'Q'|'b'|'B'

force => BOOLEAN

language => STRING

log => LEVEL

trace => LEVEL

example:

```
my $mmfu = 'Mail::Message::Field::URIs';
my $f = $mmfu->new( 'List-Post' => 'mailto:x@y.com' );
my $f = $mmfu->new( 'List-Post' => '<mailto:x@y.com>' );
my $f = $mmfu->new( 'List-Post: <mailto:x@y.com>' );
my $f = $mmfu->new( 'List-Post' => [ $uri, 'http://x.org' ] );
```

The field

Extends “The field” in Mail::Message::Field::Structured.

`$obj->isStructured()`

Mail::Message::Field::URIs->isStructured()

Inherited, see “The field” in Mail::Message::Field

`$obj->length()`

Inherited, see “The field” in Mail::Message::Field

`$obj->nrLines()`

Inherited, see “The field” in Mail::Message::Field

`$obj->print([$fh])`

Inherited, see “The field” in Mail::Message::Field

`$obj->size()`

Inherited, see “The field” in Mail::Message::Field

`$obj->string([$wrap])`

Inherited, see “The field” in Mail::Message::Field

`$obj->toDisclose()`

Inherited, see “The field” in Mail::Message::Field

Access to the name

Extends “Access to the name” in Mail::Message::Field::Structured.

`$obj->Name()`

Inherited, see “Access to the name” in Mail::Message::Field

`$obj->name()`

Inherited, see “Access to the name” in Mail::Message::Field

`$obj->wellformedName([STRING])`

Inherited, see “Access to the name” in Mail::Message::Field

Access to the body

Extends “Access to the body” in Mail::Message::Field::Structured.

`$obj->body()`

Inherited, see “Access to the body” in Mail::Message::Field

`$obj->decodedBody(%options)`

Inherited, see “Access to the body” in Mail::Message::Field::Full

`$obj->folded()`

Inherited, see “Access to the body” in Mail::Message::Field

`$obj->foldedBody([$body])`

Inherited, see “Access to the body” in Mail::Message::Field

`$obj->stripCFWS([STRING])`

Mail::Message::Field::URIs->stripCFWS([STRING])

Inherited, see “Access to the body” in Mail::Message::Field

`$obj->unfoldedBody([$body, [$wrap]])`

Inherited, see “Access to the body” in Mail::Message::Field

Access to the content

Extends “Access to the content” in Mail::Message::Field::Structured.

`$obj->URIs()`

Returns a list with all URIs defined by the field. Mind the lower-case ‘s’ at the end of the name.

example:

```
my @uris = $field->URIs;
```

`$obj->addAttribute(...)`

Attributes are not supported for URI fields.

`$obj->addURI($uri)`

Add an `$uri` to the field. The `$uri` can be specified as URI object or as string which will be turned into an `$uri` object. The added `$uri` is returned.

example: adding an URI to an URI field

```
my $f = Mail::Message::Field::URI->new( 'List-Post' );
```

```
my $uri = URI->new( "http://x.org" );
$f->addURI( $uri );
```

```
$f->addURI( "http://y.org" ); # simpler
$f->addURI( "//y.org", "http" );
```

`$obj->addresses()`

Inherited, see “Access to the content” in Mail::Message::Field

`$obj->attrPairs()`

Inherited, see “Access to the content” in Mail::Message::Field::Structured

`$obj->attribute($object|<STRING, %options>|<$name,$value,%options>)`

Inherited, see “Access to the content” in Mail::Message::Field::Structured

`$obj->attributes()`

Inherited, see “Access to the content” in Mail::Message::Field::Structured

`$obj->beautify()`

Inherited, see “Access to the content” in Mail::Message::Field::Full

`$obj->comment([STRING])`

Inherited, see “Access to the content” in Mail::Message::Field

`$obj->createComment(STRING, %options)`

Mail::Message::Field::URIs->**createComment**(STRING, %options)

Inherited, see “Access to the content” in Mail::Message::Field::Full

`$obj->createPhrase(STRING, %options)`

Mail::Message::Field::URIs->**createPhrase**(STRING, %options)

Inherited, see “Access to the content” in Mail::Message::Field::Full

`$obj->study()`

Inherited, see “Access to the content” in Mail::Message::Field

`$obj->toDate([$time])`

Mail::Message::Field::URIs->**toDate**([\$time])

Inherited, see “Access to the content” in Mail::Message::Field

`$obj->toInt()`

Inherited, see “Access to the content” in Mail::Message::Field

Other methods

Extends “Other methods” in Mail::Message::Field::Structured.

`$obj->dateToTimestamp(STRING)`

Mail::Message::Field::URIs->**dateToTimestamp**(STRING)

Inherited, see “Other methods” in Mail::Message::Field

Internals

Extends “Internals” in Mail::Message::Field::Structured.

`$obj->consume($line | <$name,<$body|$objects>>)`

Inherited, see “Internals” in Mail::Message::Field

`$obj->decode(STRING, %options)`

Mail::Message::Field::URIs->**decode**(STRING, %options)

Inherited, see “Internals” in Mail::Message::Field::Full

`$obj->defaultWrapLength([$length])`

Inherited, see “Internals” in Mail::Message::Field

`$obj->encode(STRING, %options)`

Inherited, see “Internals” in Mail::Message::Field::Full

`$obj->fold($name, $body, [$maxchars])`

Mail::Message::Field::URIs->**fold**(\$name, \$body, [\$maxchars])

Inherited, see “Internals” in Mail::Message::Field

`$obj->setWrapLength([$length])`

Inherited, see “Internals” in Mail::Message::Field

`$obj->stringifyData(STRING|ARRAY|$objects)`

Inherited, see “Internals” in Mail::Message::Field

`$obj->unfold(STRING)`

Inherited, see “Internals” in Mail::Message::Field

Parsing

Extends “Parsing” in Mail::Message::Field::Structured.

`$obj->consumeComment(STRING)`

Mail::Message::Field::URIs->**consumeComment**(STRING)

Inherited, see “Parsing” in Mail::Message::Field::Full

\$obj->**consumeDotAtom**(STRING)

Inherited, see “Parsing” in Mail::Message::Field::Full

\$obj->**consumePhrase**(STRING)

Mail::Message::Field::URIs->**consumePhrase**(STRING)

Inherited, see “Parsing” in Mail::Message::Field::Full

\$obj->**datum**([\$value])

Inherited, see “Parsing” in Mail::Message::Field::Structured

\$obj->**parse**(STRING)

Inherited, see “Parsing” in Mail::Message::Field::Full

\$obj->**produceBody**()

Inherited, see “Parsing” in Mail::Message::Field::Full

Error handling

Extends “Error handling” in Mail::Message::Field::Structured.

\$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::Field::URIs->**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::Field::URIs->**log**([\$level, [\$strings]])

Inherited, see “Error handling” in Mail::Reporter

\$obj->**logPriority**(\$level)

Mail::Message::Field::URIs->**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::Field::Structured.

`$obj->DESTROY()`

Inherited, see “Cleanup” in Mail::Reporter

DETAILS

Extends “DETAILS” in Mail::Message::Field::Structured.

DIAGNOSTICS

Warning: Field content is not numerical: `$content`

The numeric value of a field is requested (for instance the `Lines` or `Content-Length` fields should be numerical), however the data contains weird characters.

Warning: Illegal character in charset '`$charset`'

The field is created with an utf8 string which only contains data from the specified character set. However, that character set can never be a valid name because it contains characters which are not permitted.

Warning: Illegal character in field name `$name`

A new field is being created which does contain characters not permitted by the RFCs. Using this field in messages may break other e-mail clients or transfer agents, and therefore mutilate or extinguish your message.

Warning: Illegal character in language '`$lang`'

The field is created with data which is specified to be in a certain language, however, the name of the language cannot be valid: it contains characters which are not permitted by the RFCs.

Warning: Illegal encoding '`$encoding`', used '`q`'

The RFCs only permit base64 (`b` or `B`) or quoted-printable (`q` or `Q`) encoding. Other than these four options are illegal.

Error: No attributes for URI fields.

Is is not possible to add attributes to URI fields: it is not permitted by the RFCs.

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

Copyrights 2001–2022 by [Mark Overmeer <markov@cpan.org>]. For other contributors see ChangeLog.

This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself. See <http://dev.perl.org/licenses/>