



**Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!**

***Rocky Enterprise Linux 9.2 Manual Pages on command 'WWW::Form::UrlEncoded.3pm'***

***\$ man WWW::Form::UrlEncoded.3pm***

WWW::Form::UrlEncoded(3pm) User Contributed Perl Documentation WWW::Form::UrlEncoded(3pm)

**NAME**

WWW::Form::UrlEncoded - parser and builder for application/x-www-form-urlencoded

**SYNOPSIS**

```
use WWW::Form::UrlEncoded qw/parse_urlencoded build_urlencoded/;
```

```
my $query_string = "foo=bar&baz=param";
```

```
my @params = parse_urlencoded($query_string);
```

```
# ('foo','bar','baz','param')
```

```
my $query_string = build_urlencoded('foo','bar','baz','param');
```

```
# "foo=bar&baz=param";
```

**DESCRIPTION**

WWW::Form::UrlEncoded provides application/x-www-form-urlencoded parser and builder. This module aims to have compatibility with other CPAN modules like HTTP::Body's urlencoded parser.

This module try to use WWW::Form::UrlEncoded::XS by default and fail to it, use

WWW::Form::UrlEncoded::PP instead

## Parser rules

WWW::Form::UrlEncoded parsed string in this rule.

1. Split application/x-www-form-urlencoded payload by "&" (U+0026) or ";" (U+003B)
2. Ready empty array to store "name" and "value"
3. For each divided string, apply next steps.
  1. If first character of string is ' ' (U+0020 SPACE), remove it.
  2. If string has "=", let name be substring from start to first "=", but excluding first "=", and remains to be value. If there is no strings after first "=", value to be empty string "". If first "=" is first character of the string, let key be empty string "". If string does not have any "=", all of the string to be key and value to be empty string "".
  3. replace all "+" (U+002B) with ' ' (U+0020 SPACE).
  4. unescape name and value. push them to the array.
4. return the array.

## Test data

'a=b&c=d' => ["a","b","c","d"]

'a=b;c=d' => ["a","b","c","d"]

'a=1&b=2;c=3' => ["a","1","b","2","c","3"]

'a==b&c==d' => ["a","=b","c","=d"]

'a=b& c=d' => ["a","b","c","d"]

'a=b; c=d' => ["a","b","c","d"]

'a=b; c =d' => ["a","b","c ","d"]

'a=b;c= d ' => ["a","b","c"," d "]

'a=b&+c=d' => ["a","b"," c","d"]

'a=b&+c+=d' => ["a","b"," c ","d"]

'a=b&c+=d+' => ["a","b","c"," d "]

'a=b&%20c=d' => ["a","b"," c","d"]

'a=b&%20c%20=d' => ["a","b"," c ","d"]

'a=b&c=%20d%20' => ["a","b","c"," d "]

'a&c=d' => ["a","","c","d"]

'a=b&=d' => ["a","b","","d"]

'a=b&=' => ["a","b","",""]

'&' => [","","",""]

'=' => [","",""]

" => []

## FUNCTION

@param = parse\_urlencoded(\$str:String)

parse \$str and return Array that contains key-value pairs.

\$param:ArrayRef = parse\_urlencoded\_arrayref(\$str:String)

parse \$str and return ArrayRef that contains key-value pairs.

\$string = build\_urlencoded(@param)

\$string = build\_urlencoded(@param, \$delim)

\$string = build\_urlencoded(\@param)

\$string = build\_urlencoded(\@param, \$delim)

\$string = build\_urlencoded(\%param)

\$string = build\_urlencoded(\%param, \$delim)

build urlencoded string from param. build\_urlencoded accepts arrayref and hashref

values.

```
build_urlencoded( foo => 1, foo => 2);
```

```
build_urlencoded( foo => [1,2] );
```

```
build_urlencoded( [ foo => 1, foo => 2 ] );
```

```
build_urlencoded( [foo => [1,2]] );
```

```
build_urlencoded( {foo => [1,2]} );
```

If \$delim parameter is passed, this function use it instead of using "&".

\$string = build\_urlencoded\_utf8(...)

This function is almost same as "build\_urlencoded". build\_urlencoded\_utf8 call

"utf8::encode" for all parameters.

## ENVIRONMENT VALUE

? WWW\_FORM\_URL ENCODED\_PP

If true, WWW::Form::UrlEncoded force to load WWW::Form::UrlEncoded::PP.

## SEE ALSO

CPAN already has some application/x-www-form-urlencoded parser modules like these.

URL::Encode

URL::Encode::XS

Text::QueryString

They does not fully compatible with WWW::Form::UrlEncoded. Handling of empty key-value and supporting separator characters are different.

## LICENSE

Copyright (C) Masahiro Nagano.

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself.

## AUTHOR

Masahiro Nagano <kazeburo@gmail.com>

perl v5.30.0

2019-11-10

WWW::Form::UrlEncoded(3pm)