# encodeurl

[NPM Version](https://npmjs.org/package/encodeurl) [NPM Downloads](https://npmjs.org/package/encodeurl) [Node.js Version](https://nodejs.org/en/download) [Build Status](https://travis-ci.org/pillarjs/encodeurl) [Test Coverage](https://coveralls.io/r/pillarjs/encodeurl?branch=master)

Encode a URL to a percent-encoded form, excluding already-encoded sequences

## Installation

This is a [Node.js](https://nodejs.org/en/) module available through the [npm registry](https://www.npmjs.com/). Installation is done using the [npm install command](https://docs.npmjs.com/getting-started/installing-npm-packages-locally):

$ npm install encodeurl

## API

var encodeUrl = require('encodeurl')

### encodeUrl(url)

Encode a URL to a percent-encoded form, excluding already-encoded sequences.

This function will take an already-encoded URL and encode all the non-URL code points (as UTF-8 byte sequences). This function will not encode the "%" character unless it is not part of a valid sequence (%20 will be left as-is, but %foo will be encoded as %25foo).

This encode is meant to be "safe" and does not throw errors. It will try as hard as it can to properly encode the given URL, including replacing any raw, unpaired surrogate pairs with the Unicode replacement character prior to encoding.

This function is *similar* to the intrinsic function encodeURI, except it will not encode the % character if that is part of a valid sequence, will not encode [ and ] (for IPv6 hostnames) and will replace raw, unpaired surrogate pairs with the Unicode replacement character (instead of throwing).

## Examples

### Encode a URL containing user-controled data

var encodeUrl = require('encodeurl')

var escapeHtml = require('escape-html')

http.createServer(function onRequest (req, res) {

// get encoded form of inbound url

var url = encodeUrl(req.url)

// create html message

var body = '<p>Location ' + escapeHtml(url) + ' not found</p>'

// send a 404

res.statusCode = 404

res.setHeader('Content-Type', 'text/html; charset=UTF-8')

res.setHeader('Content-Length', String(Buffer.byteLength(body, 'utf-8')))

res.end(body, 'utf-8')

})

### Encode a URL for use in a header field

var encodeUrl = require('encodeurl')

var escapeHtml = require('escape-html')

var url = require('url')

http.createServer(function onRequest (req, res) {

// parse inbound url

var href = url.parse(req)

// set new host for redirect

href.host = 'localhost'

href.protocol = 'https:'

href.slashes = true

// create location header

var location = encodeUrl(url.format(href))

// create html message

var body = '<p>Redirecting to new site: ' + escapeHtml(location) + '</p>'

// send a 301

res.statusCode = 301

res.setHeader('Content-Type', 'text/html; charset=UTF-8')

res.setHeader('Content-Length', String(Buffer.byteLength(body, 'utf-8')))

res.setHeader('Location', location)

res.end(body, 'utf-8')

})

## Testing

$ npm test

$ npm run lint

## References

* [RFC 3986: Uniform Resource Identifier (URI): Generic Syntax](https://tools.ietf.org/html/rfc3986)
* [WHATWG URL Living Standard](https://url.spec.whatwg.org/)

## License

[MIT](http://license)