# content-disposition

[NPM Version](https://npmjs.org/package/content-disposition) [NPM Downloads](https://npmjs.org/package/content-disposition) [Node.js Version](https://nodejs.org/en/download) [Build Status](https://github.com/jshttp/content-disposition?query=workflow%3Aci) [Test Coverage](https://coveralls.io/r/jshttp/content-disposition?branch=master)

Create and parse HTTP Content-Disposition header

## Installation

$ npm install content-disposition

## API

var contentDisposition = require('content-disposition')

### contentDisposition(filename, options)

Create an attachment Content-Disposition header value using the given file name, if supplied. The filename is optional and if no file name is desired, but you want to specify options, set filename to undefined.

res.setHeader('Content-Disposition', contentDisposition('∫ maths.pdf'))

**note** HTTP headers are of the ISO-8859-1 character set. If you are writing this header through a means different from setHeader in Node.js, you'll want to specify the 'binary' encoding in Node.js.

#### Options

contentDisposition accepts these properties in the options object.

##### fallback

If the filename option is outside ISO-8859-1, then the file name is actually stored in a supplemental field for clients that support Unicode file names and a ISO-8859-1 version of the file name is automatically generated.

This specifies the ISO-8859-1 file name to override the automatic generation or disables the generation all together, defaults to true.

* A string will specify the ISO-8859-1 file name to use in place of automatic generation.
* false will disable including a ISO-8859-1 file name and only include the Unicode version (unless the file name is already ISO-8859-1).
* true will enable automatic generation if the file name is outside ISO-8859-1.

If the filename option is ISO-8859-1 and this option is specified and has a different value, then the filename option is encoded in the extended field and this set as the fallback field, even though they are both ISO-8859-1.

##### type

Specifies the disposition type, defaults to "attachment". This can also be "inline", or any other value (all values except inline are treated like attachment, but can convey additional information if both parties agree to it). The type is normalized to lower-case.

### contentDisposition.parse(string)

var disposition = contentDisposition.parse('attachment; filename="EURO rates.txt"; filename\*=UTF-8\'\'%e2%82%ac%20rates.txt')

Parse a Content-Disposition header string. This automatically handles extended ("Unicode") parameters by decoding them and providing them under the standard parameter name. This will return an object with the following properties (examples are shown for the string 'attachment; filename="EURO rates.txt"; filename\*=UTF-8\'\'%e2%82%ac%20rates.txt'):

* type: The disposition type (always lower case). Example: 'attachment'
* parameters: An object of the parameters in the disposition (name of parameter always lower case and extended versions replace non-extended versions). Example: {filename: "€ rates.txt"}

## Examples

### Send a file for download

var contentDisposition = require('content-disposition')

var destroy = require('destroy')

var fs = require('fs')

var http = require('http')

var onFinished = require('on-finished')

var filePath = '/path/to/public/plans.pdf'

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

// set headers

res.setHeader('Content-Type', 'application/pdf')

res.setHeader('Content-Disposition', contentDisposition(filePath))

// send file

var stream = fs.createReadStream(filePath)

stream.pipe(res)

onFinished(res, function () {

destroy(stream)

})

})

## Testing

$ npm test

## References

* [RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1](https://tools.ietf.org/html/rfc2616)
* [RFC 5987: Character Set and Language Encoding for Hypertext Transfer Protocol (HTTP) Header Field Parameters](https://tools.ietf.org/html/rfc5987)
* [RFC 6266: Use of the Content-Disposition Header Field in the Hypertext Transfer Protocol (HTTP)](https://tools.ietf.org/html/rfc6266)
* [Test Cases for HTTP Content-Disposition header field (RFC 6266) and the Encodings defined in RFCs 2047, 2231 and 5987](http://greenbytes.de/tech/tc2231/)

## License

[MIT](http://license)