# content-type

[NPM Version](https://npmjs.org/package/content-type) [NPM Downloads](https://npmjs.org/package/content-type) [Node.js Version](https://nodejs.org/en/download) [Build Status](https://github.com/jshttp/content-type/actions/workflows/ci.yml) [Coverage Status](https://coveralls.io/r/jshttp/content-type?branch=master)

Create and parse HTTP Content-Type header according to RFC 7231

## Installation

$ npm install content-type

## API

var contentType = require('content-type')

### contentType.parse(string)

var obj = contentType.parse('image/svg+xml; charset=utf-8')

Parse a Content-Type header. This will return an object with the following properties (examples are shown for the string 'image/svg+xml; charset=utf-8'):

* type: The media type (the type and subtype, always lower case). Example: 'image/svg+xml'
* parameters: An object of the parameters in the media type (name of parameter always lower case). Example: {charset: 'utf-8'}

Throws a TypeError if the string is missing or invalid.

### contentType.parse(req)

var obj = contentType.parse(req)

Parse the Content-Type header from the given req. Short-cut for contentType.parse(req.headers['content-type']).

Throws a TypeError if the Content-Type header is missing or invalid.

### contentType.parse(res)

var obj = contentType.parse(res)

Parse the Content-Type header set on the given res. Short-cut for contentType.parse(res.getHeader('content-type')).

Throws a TypeError if the Content-Type header is missing or invalid.

### contentType.format(obj)

var str = contentType.format({

type: 'image/svg+xml',

parameters: { charset: 'utf-8' }

})

Format an object into a Content-Type header. This will return a string of the content type for the given object with the following properties (examples are shown that produce the string 'image/svg+xml; charset=utf-8'):

* type: The media type (will be lower-cased). Example: 'image/svg+xml'
* parameters: An object of the parameters in the media type (name of the parameter will be lower-cased). Example: {charset: 'utf-8'}

Throws a TypeError if the object contains an invalid type or parameter names.

## License

[MIT](http://license)