

- pathname

Difference with node's url:

1. No leading slash in paths, e.g. in `url.parse('http://foo?bar')` pathname is `'',` not `'/'`
2. Backslashes are not replaced with slashes, so `http:\example.org\` is treated like a relative path
3. Trailing colon is treated like a part of the path, i.e. in `http://example.org:foo` pathname is `:foo`
4. Nothing is URL-encoded in the resulting object, (in joyent/node some chars in auth and paths are encoded)
5. `url.parse()` does not have `parseQueryString` argument
6. Removed extraneous result properties: `host`, `path`, `query`, etc., which can be constructed using other parts of the url.

.format(urlObject)

Format an object previously obtained with `.parse()` function. Similar to node's [url.format](#).

License

[MIT](#)

mdurl



URL utilities for [markdown-it](#) parser.

API

.encode(str [, exclude, keepEncoded]) -> String

Percent-encode a string, avoiding double encoding. Don't touch `/a-zA-Z0-9/` + excluded chars + `/%[a-zA-Z0-9]{2}/` (if not disabled). Broken surrogates are replaced with `U+FFFD`.

Params:

str - input string.

exclude - optional, `;/?:@&=+$,-_!.~*'()#`. Additional chars to keep intact (except `/a-zA-Z0-9/`).

keepEncoded - optional, `true`. By default it skips already encoded sequences `(/%[a-zA-Z0-9]{2}/)`. If set to `false`, % will be encoded.

encode.defaultChars, encode.componentChars

You can use these constants as second argument to encode

function.

- `encode.defaultChars` is the same exclude set as in the `standard encodeURI()` function
- `encode.componentChars` is the same exclude set as in

the `encodeURIComponent()` function

For example, `encode('something')`,

`encode.componentChars, true`) is roughly the equivalent of the `encodeURIComponent()` function (except `encode()` doesn't throw).

.decode(str [, exclude]) -> String

Decode percent-encoded string. Invalid percent-encoded sequences (e.g. %2G) are left as is. Invalid UTF-8 characters are replaced with U+FFFD.

Params:

- **str** - input string.
- **exclude** - set of characters to leave encoded, optional, `;/?:@&=+$, #`.

decode.defaultChars, decode.componentChars

You can use these constants as second argument to decode

function.

- `decode.defaultChars` is the same exclude set as in the `standard decodeURI()` function
- `decode.componentChars` is the same exclude set as in

the `decodeURIComponent()` function

For example, `decode('something')`, `decode.defaultChars`) has the same behavior as `decodeURI('something')` on a correctly encoded input.

.parse(url, slashesDenoteHost) -> urlObs

Parse url string. Similar to node's [url.parse](#), but without any normalizations and query string parse.

- **url** - input url (string)
- **slashesDenoteHost** - if url starts with `//`, expect a hostname after it. Optional, `false`.

Result (hash):

- protocol
- slashes
- auth
- port
- hostname
- hash
- search