/\*\*

\* The `querystring` module provides utilities for parsing and formatting URL

\* query strings. It can be accessed using:

\*

\* ```js

\* const querystring = require('querystring');

\* ```

\*

\* The `querystring` API is considered Legacy. While it is still maintained,

\* new code should use the `URLSearchParams` API instead.

\* @deprecated Legacy

\* @see [source](https://github.com/nodejs/node/blob/v17.0.0/lib/querystring.js)

\*/

declare module 'querystring' {

interface StringifyOptions {

encodeURIComponent?: ((str: string) => string) | undefined;

}

interface ParseOptions {

maxKeys?: number | undefined;

decodeURIComponent?: ((str: string) => string) | undefined;

}

interface ParsedUrlQuery extends NodeJS.Dict<string | string[]> {}

interface ParsedUrlQueryInput extends NodeJS.Dict<string | number | boolean | ReadonlyArray<string> | ReadonlyArray<number> | ReadonlyArray<boolean> | null> {}

/\*\*

\* The `querystring.stringify()` method produces a URL query string from a

\* given `obj` by iterating through the object's "own properties".

\*

\* It serializes the following types of values passed in `obj`:[string](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data\_structures#String\_type) |

\* [number](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data\_structures#Number\_type) |

\* [bigint](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/BigInt) |

\* [boolean](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data\_structures#Boolean\_type) |

\* [string\[\]](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data\_structures#String\_type) |

\* [number\[\]](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data\_structures#Number\_type) |

\* [bigint\[\]](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/BigInt) |

\* [boolean\[\]](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data\_structures#Boolean\_type) The numeric values must be finite. Any other input values will be coerced to

\* empty strings.

\*

\* ```js

\* querystring.stringify({ foo: 'bar', baz: ['qux', 'quux'], corge: '' });

\* // Returns 'foo=bar&#x26;baz=qux&#x26;baz=quux&#x26;corge='

\*

\* querystring.stringify({ foo: 'bar', baz: 'qux' }, ';', ':');

\* // Returns 'foo:bar;baz:qux'

\* ```

\*

\* By default, characters requiring percent-encoding within the query string will

\* be encoded as UTF-8\. If an alternative encoding is required, then an alternative`encodeURIComponent` option will need to be specified:

\*

\* ```js

\* // Assuming gbkEncodeURIComponent function already exists,

\*

\* querystring.stringify({ w: '中文', foo: 'bar' }, null, null,

\* { encodeURIComponent: gbkEncodeURIComponent });

\* ```

\* @since v0.1.25

\* @param obj The object to serialize into a URL query string

\* @param [sep='&'] The substring used to delimit key and value pairs in the query string.

\* @param [eq='='] . The substring used to delimit keys and values in the query string.

\*/

function stringify(obj?: ParsedUrlQueryInput, sep?: string, eq?: string, options?: StringifyOptions): string;

/\*\*

\* The `querystring.parse()` method parses a URL query string (`str`) into a

\* collection of key and value pairs.

\*

\* For example, the query string `'foo=bar&#x26;abc=xyz&#x26;abc=123'` is parsed into:

\*

\* ```js

\* {

\* foo: 'bar',

\* abc: ['xyz', '123']

\* }

\* ```

\*

\* The object returned by the `querystring.parse()` method \_does not\_prototypically inherit from the JavaScript `Object`. This means that typical`Object` methods such as `obj.toString()`,

\* `obj.hasOwnProperty()`, and others

\* are not defined and \_will not work\_.

\*

\* By default, percent-encoded characters within the query string will be assumed

\* to use UTF-8 encoding. If an alternative character encoding is used, then an

\* alternative `decodeURIComponent` option will need to be specified:

\*

\* ```js

\* // Assuming gbkDecodeURIComponent function already exists...

\*

\* querystring.parse('w=%D6%D0%CE%C4&#x26;foo=bar', null, null,

\* { decodeURIComponent: gbkDecodeURIComponent });

\* ```

\* @since v0.1.25

\* @param str The URL query string to parse

\* @param [sep='&'] The substring used to delimit key and value pairs in the query string.

\* @param [eq='='] . The substring used to delimit keys and values in the query string.

\*/

function parse(str: string, sep?: string, eq?: string, options?: ParseOptions): ParsedUrlQuery;

/\*\*

\* The querystring.encode() function is an alias for querystring.stringify().

\*/

const encode: typeof stringify;

/\*\*

\* The querystring.decode() function is an alias for querystring.parse().

\*/

const decode: typeof parse;

/\*\*

\* The `querystring.escape()` method performs URL percent-encoding on the given`str` in a manner that is optimized for the specific requirements of URL

\* query strings.

\*

\* The `querystring.escape()` method is used by `querystring.stringify()` and is

\* generally not expected to be used directly. It is exported primarily to allow

\* application code to provide a replacement percent-encoding implementation if

\* necessary by assigning `querystring.escape` to an alternative function.

\* @since v0.1.25

\*/

function escape(str: string): string;

/\*\*

\* The `querystring.unescape()` method performs decoding of URL percent-encoded

\* characters on the given `str`.

\*

\* The `querystring.unescape()` method is used by `querystring.parse()` and is

\* generally not expected to be used directly. It is exported primarily to allow

\* application code to provide a replacement decoding implementation if

\* necessary by assigning `querystring.unescape` to an alternative function.

\*

\* By default, the `querystring.unescape()` method will attempt to use the

\* JavaScript built-in `decodeURIComponent()` method to decode. If that fails,

\* a safer equivalent that does not throw on malformed URLs will be used.

\* @since v0.1.25

\*/

function unescape(str: string): string;

}

declare module 'node:querystring' {

export \* from 'querystring';

}