util

> util

- [Doc] URL
- [Doc] Query Strings ( )
- [Doc] Utilities ( )
- [Basic]

URL

```
┌──────────────────────────────────────────────────────────────────────────────┐
│                                    href                                      │
├──────────┬┬───────────┬──────────────────┬────────────────────────────┬───────┤
│ protocol ||    auth   │       host       │            path            │ hash  │
│          ||           ├──────────┬───────┼────────────┬───────────────┤       │
│          ||           │ hostname │ port  │ pathname   │     search    │       │
│          ||           │          │       │            ├─┬─────────────┤       │
│          ||           │          │       │            │ │    query    │       │
"  http:    // user:pass @ host.com : 8080    /p/a/t/h  ?  query=string   #hash "
│          ||           │          │       │            │ │             │       │
└──────────┴┴───────────┴──────────┴───────┴────────────┴─┴─────────────┴───────┘
```

:

	encodeURI
[' ']	['%20']
[<]	['%3C']
[>]	['%3E']
["]	['%22']
``	['%60']
[\r]	['%0D']
[\n]	['%0A']
[\t]	['%09']
[{]	['%7B']

	encodeURI
`}`	`'%7D'`
`	`
`\\`	`'%5C'`
`^`	`'%5E'`
`'`	'%27'

? :

```
Array(range).fill(0)
  .map((_, i) => String.fromCharCode(i))
  .map(encodeURI)
```

range 255 (doge

Query Strings

query string URL , URL `querystring` .

.parse(str[, sep[, eq[, options]]])	query string json
.unescape(str)	.parse ,
.stringify(obj[, sep[, eq[, options]]])	json query string
.escape(str)	.stringify ,

Node.js querystring . :

```
const qs = require('qs'); //
const querystring = require('querystring'); // Node.js

let obj = { a: { b: { c: 1 } } };

console.log(qs.stringify(obj)); // 'a%5Bb%5D%5Bc%5D=1'
console.log(querystring.stringify(obj)); // 'a='

let str = 'a%5Bb%5D%5Bc%5D=1';

console.log(qs.parse(str)); // { a: { b: { c: '1' } } }
console.log(querystring.parse(str)); // { 'a[b][c]': '1' }
```

> HTTP GET (URL) let arr = [1,2,3,4] ?

```
const qs = require('qs');

let arr = [1,2,3,4];
let str = qs.stringify({arr});

console.log(str); // arr%5B0%5D=1&arr%5B1%5D=2&arr%5B2%5D=3&arr%5B3%5D=4
console.log(decodeURI(str)); // 'arr[0]=1&arr[1]=2&arr[2]=3&arr[3]=4'
console.log(qs.parse(str)); // { arr: [ '1', '2', '3', '4' ] }
```

`https://your.host/api/?arr[0]=1&arr[1]=2&arr[2]=3&arr[3]=4` arr

util

util.is*() v4.0.0 (deprecated). , , .

- util.debug(string)
- util.error([...strings])
- util.isArray(object)
- util.isBoolean(object)
- util.isBuffer(object)
- util.isDate(object)
- util.isError(object)
- util.isFunction(object)
- util.isNull(object)

- util.isNullOrUndefined(object)
- util.isNumber(object)
- util.isObject(object)
- util.isPrimitive(object)
- util.isRegExp(object)
- util.isString(object)
- util.isSymbol(object)
- util.isUndefined(object)
- util.log(string)
- util.print([...strings])
- util.puts([...strings])
- util._extend(target, source)

.

util.inherits

> Node.js (util.inherits) ?

https://github.com/nodejs/node/blob/v7.6.0/lib/util.js#L960

```
/**
 * Inherit the prototype methods from one constructor into another.
 *
 * The Function.prototype.inherits from lang.js rewritten as a standalone
 * function (not on Function.prototype). NOTE: If this file is to be loaded
 * during bootstrapping this function needs to be rewritten using some native
 * functions as prototype setup using normal JavaScript does not work as
 * expected during bootstrapping (see mirror.js in r114903).
 *
 * @param {function} ctor Constructor function which needs to inherit the
 *     prototype.
 * @param {function} superCtor Constructor function to inherit prototype from.
 * @throws {TypeError} Will error if either constructor is null, or if
 *     the super constructor lacks a prototype.
 */
exports.inherits = function(ctor, superCtor) {

  if (ctor === undefined || ctor === null)
    throw new TypeError('The constructor to "inherits" must not be ' +
```

```
                        'null or undefined');

  if (superCtor === undefined || superCtor === null)
    throw new TypeError('The super constructor to "inherits" must not '
+
                        'be null or undefined');

  if (superCtor.prototype === undefined)
    throw new TypeError('The super constructor to "inherits" must ' +
                        'have a prototype');

  ctor.super_ = superCtor;
  Object.setPrototypeOf(ctor.prototype, superCtor.prototype);
};
```

, GNU .

..

Awesome Node.js Most depended-upon packages

> “ ?

:

```
const fs = require('fs');
const path = require('path');

function traversal(dir) {
  let res = []
  for (let item of fs.readdirSync(dir)) {
    let filepath = path.join(dir, item);
    try {
      let fd = fs.openSync(filepath, 'r');
      let flag = fs.fstatSync(fd).isDirectory();
      fs.close(fd); // TODO
      if (flag) {
        res.push(...traversal(filepath));
```

```
      } else {
        res.push(filepath);
      }
    } catch(err) {
      if (err.code === 'ENOENT' && // link
          !!fs.readlinkSync(filepath)) { //     link
        res.push(filepath);
      } else {
        console.error('err', err);
      }
    }
  }
  return res.map((file) => path.basename(file));
}

console.log(traversal('.'));
```

Oh only:

```
const glob = require("glob");

glob("**/*.js", (err, files) => {
  if (err) {
    throw new Error(err);
  }
  files.map((filename) => {
    console.log('Here you are:', filename);
  });
});
```

Revision #1
Created 3 August 2021 02:19:37 by
Updated 3 August 2021 02:20:47 by