

目录

- `querystring` (查询字符串)
  - `querystring.escape(str)`
  - `querystring.parse(str[, sep[, eq[, options]]])`
  - `querystring.stringify(obj[, sep[, eq[, options]]])`
  - `querystring.unescape(str)`

# querystring (查询字符串) #

[查看英文版](#) / [参与翻译](#)

Stability: 2 – Stable

`querystring` 模块提供了一些实用工具，用于解析与格式化 URL 查询字符串。它可以通过以下方式被使用：

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

## querystring.escape(str) #

[查看英文版](#) / [参与翻译](#)

新增于: v0.1.25

- `str` `<String>`

`querystring.escape()` 方法对给定的 `str` 执行 URL 百分号编码。

`querystring.escape()` 方法是供 `querystring.stringify()` 使用的，且通常不被直接使用。它之所以对外开放，是为了在需要时可以通过给 `querystring.escape` 赋值一个函数来重写编码的实现。

## querystring.parse(str[, sep[, eq[, options]]]) #

[查看英文版](#) / [参与翻译](#)

新增于: v0.1.25

- `str` `<String>` 要解析的 URL 查询字符串。
- `sep` `<String>` 用于界定查询字符串中的键值对的子字符串。默认为 `'&'`。
- `eq` `<String>` 用于界定查询字符串中的键与值的子字符串。默认为 `'='`。
- `options` `<Object>`
  - `decodeURIComponent` `<Function>` 当解码查询字符串中百分号编码的字符时使用的函数。默认为 `querystring.unescape()`。
  - `maxKeys` `<number>` 指定要解析的键的最大数量。默认为 `1000`。指定为 `0` 则移除键数的限制。

`querystring.parse()` 方法能把一个 URL 查询字符串（`str`）解析成一个键值对的集合。

例子，查询字符串 `'foo=bar&abc=xyz&abc=123'` 被解析成：

```
{
  foo: 'bar',
  abc: ['xyz', '123']
}
```

注意：`querystring.parse()` 方法返回的对象不继承自 JavaScript 的 `Object`。这意味着典型的 `Object` 方法如 `obj.toString()`、`obj.hasOwnProperty()` 等没有被定义且无法使用。

默认情况下，查询字符串中的百分号编码的字符会被认为使用了 UTF-8 编码。如果使用的是另一种字符编码，则 `decodeURIComponent` 选项需要被指定，如下例子：

```
// 假设 gbkDecodeURIComponent 函数已存在。

querystring.parse('w=%D6%D0%CE%C4&foo=bar', null, null,
  { decodeURIComponent: gbkDecodeURIComponent })
```

## querystring.stringify(obj[, sep[, eq[, options]]]) #

[查看英文版](#) / [参与翻译](#)

新增于: v0.1.25

- `obj` `<Object>` 要序列化成一个 URL 查询字符串的对象。
- `sep` `<String>` 用于界定查询字符串中的键值对的子字符串。默认为 `'&'`。
- `eq` `<String>` 用于界定查询字符串中的键与值的子字符串。默认为 `'='`。
- `options`
  - `encodeURIComponent` `<Function>` 当把对 URL 不安全的字符转换成查询字符串中的百分号编码时使用的函数。默认为 `querystring.escape()`。

`querystring.stringify()` 方法通过遍历对象的自有属性，从一个给定的 `obj` 产生一个 URL 查询字符串。

例子：

```
querystring.stringify({ foo: 'bar', baz: ['qux', 'quux'], corge: '' })
// 返回 'foo=bar&baz=qux&baz=quux&corge='

querystring.stringify({foo: 'bar', baz: 'qux'}, ';', ':')
// 返回 'foo:bar;baz:qux'
```

默认情况下，查询字符串中需要百分号编码的字符会作为 UTF-8 被编码。如果需要的是另一种编码，则 `encodeURIComponent` 选项需要被指定，如下例子：

```
// 假设 gbkEncodeURIComponent 函数已存在。

querystring.stringify({ w: '中文', foo: 'bar' }, null, null,
  { encodeURIComponent: gbkEncodeURIComponent })
```

## querystring.unescape(str) #

[查看英文版](#) / [参与翻译](#)

新增于: v0.1.25

- `str` `<String>`

`querystring.unescape()` 方法对给定的 `str` 上的 URL 百分号编码的字符执行解码。

`querystring.unescape()` 方法是供 `querystring.parse()` 使用的，且通常不被直接使用。它之所以对外开放，是为了在需要时可以通过给 `querystring.unescape` 赋值一个函数来重写解码的实现。

`querystring.unescape()` 方法默认使用 JavaScript 内置的 `decodeURIComponent()` 方法来解码。