

# Node.js v6.10.0 文档

[返回文档首页](#)

## 目录

- [V8](#)
  - [v8.getHeapStatistics\(\)](#)
  - [v8.getHeapSpaceStatistics\(\)](#)
  - [v8.setFlagsFromString\(string\)](#)

# V8

#

[查看英文版](#) / [参与翻译](#)

The `v8` module exposes APIs that are specific to the version of `V8` built into the Node.js binary. It can be accessed using:

```
const v8 = require('v8');
```

*Note:* The APIs and implementation are subject to change at any time.

## v8.getHeapStatistics()

#

[查看英文版](#) / [参与翻译](#)

新增于: v1.0.0

Returns an object with the following properties:

- `total_heap_size` <number>
- `total_heap_size_executable` <number>
- `total_physical_size` <number>
- `total_available_size` <number>
- `used_heap_size` <number>
- `heap_size_limit` <number>

For example:

```
{
  total_heap_size: 7326976,
  total_heap_size_executable: 4194384,
  total_physical_size: 7326976,
  total_available_size: 1152656,
  used_heap_size: 3476208,
  heap_size_limit: 1535115264
}
```

## v8.getHeapSpaceStatistics()

#

[查看英文版](#) / [参与翻译](#)

新增于: v6.0.0

Returns statistics about the V8 heap spaces, i.e. the segments which make up the V8 heap. Neither the ordering of heap spaces, nor the availability of a heap space can be guaranteed as the statistics are provided via the `V8 GetHeapSpaceStatistics` function and may change from one V8 version to the next.

The value returned is an array of objects containing the following properties:

- `space_name` <string>
- `space_size` <number>
- `space_used_size` <number>
- `space_available_size` <number>
- `physical_space_size` <number>

For example:

```
[
  {
    "space_name": "new_space",
    "space_size": 2063872,
    "space_used_size": 951112,
    "space_available_size": 80824,
    "physical_space_size": 2063872
  },
  {
    "space_name": "old_space",
    "space_size": 3090560,
    "space_used_size": 2493792,
    "space_available_size": 0,
    "physical_space_size": 3090560
  },
  {
    "space_name": "code_space",
    "space_size": 1260160,
    "space_used_size": 644256,
    "space_available_size": 960,
    "physical_space_size": 1260160
  },
  {
    "space_name": "map_space",
    "space_size": 1094160,
    "space_used_size": 201608,
    "space_available_size": 0,
    "physical_space_size": 1094160
  },
  {
    "space_name": "large_object_space",
    "space_size": 0,
    "space_used_size": 0,
    "space_available_size": 1490980608,
    "physical_space_size": 0
  }
]
```

## v8.setFlagsFromString(string)

#

[查看英文版](#) / [参与翻译](#)

新增于: v1.0.0

The `v8.setFlagsFromString()` method can be used to programmatically set V8 command line flags. This method should be used with care. Changing settings after the VM has started may result in unpredictable behavior, including crashes and data loss; or it may simply do nothing.

The V8 options available for a version of Node.js may be determined by running `node --v8-options`. An unofficial, community-maintained list of options and their effects is available [here](#).

Usage:

```
// Print GC events to stdout for one minute.
const v8 = require('v8');
v8.setFlagsFromString('--trace_gc');
setTimeout(function() { v8.setFlagsFromString('--notrace_gc'); }, 60e3);
```