Node.js v0.8.16 Manual & Documentation

Table of Contents

Child Process

```
Stability: 3 - Stable
```

Node provides a tri-directional `popen(3)` facility through the `child_process` module.

It is possible to stream data through a child's `stdin`, `stdout`, and `stderr` in a fully non-blocking way.

To create a child process use `require('child_process').spawn()` or `require('child_process').fork()`. The semantics of each are slightly different, and explained below.

Class: ChildProcess

`ChildProcess` is an EventEmitter.

Child processes always have three streams associated with them. `child.stdin`, `child.stdout`, and `child.stderr`. These may be shared with the stdio streams of the parent process, or they may be separate stream objects which can be piped to and from.

The ChildProcess class is not intended to be used directly. Use the `spawn()` or `fork()` methods to create a Child Process instance.

Event: 'exit'

- `code` Number the exit code, if it exited normally.
- `signal` String the signal passed to kill the child process, if it was killed by the parent.

This event is emitted after the child process ends. If the process terminated normally, `code` is the final exit code of the process, otherwise `null`. If the process terminated due to receipt of a signal, `signal` is the string name of the signal, otherwise `null`.

Note that the child process stdio streams might still be open.

See `waitpid(2)`.

Event: 'close'

This event is emitted when the stdio streams of a child process have all terminated. This is distinct from 'exit', since multiple processes might share the same stdio streams.

Event: 'disconnect'

This event is emitted after using the `.disconnect()` method in the parent or in the child. After disconnecting it is no longer possible to send messages. An alternative way to check if you can send messages is to see if the `child.connected` property is `true`.

Event: 'message'

- `message` Object a parsed JSON object or primitive value
- `sendHandle` Handle object a Socket or Server object

Messages send by `.send(message, [sendHandle])` are obtained using the `message` event.

child.stdin

- Stream object

A `Writable Stream` that represents the child process's `stdin`. Closing this stream via `end()` often causes the child process to terminate.

If the child stdio streams are shared with the parent, then this will not be set.

child.stdout

- Stream object

A `Readable Stream` that represents the child process's `stdout`.

If the child stdio streams are shared with the parent, then this will not be set.

child.stderr

- Stream object