

# node-cron - npm

 [npmjs.com/package/node-cron](https://npmjs.com/package/node-cron)

## Node Cron

The node-cron module is tiny task scheduler in pure JavaScript for node.js based on [GNU crontab](#). This module allows you to schedule task in node.js using full crontab syntax.

**Need a job scheduler with support for worker threads and cron syntax?** Try out the [Bree](#) job scheduler!

## Getting Started

Install node-cron using npm:

```
npm install --save node-cron
```

Import node-cron and schedule a task:

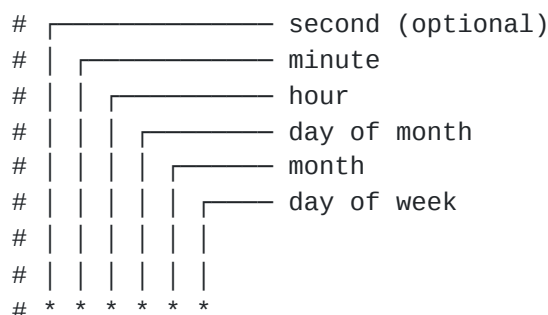
```
var cron = require('node-cron');

cron.schedule('* * * * *', () => {
  console.log('running a task every minute');
});
```

## Cron Syntax

This is a quick reference to cron syntax and also shows the options supported by node-cron.

## Allowed fields



```
# |----- second (optional)
# |----- minute
# |----- hour
# |----- day of month
# |----- month
# |----- day of week
# |
# |
# |
# * * * * *
```

## Allowed values

field	value
second	0-59

field	value
minute	0-59
hour	0-23
day of month	1-31
month	1-12 (or names)
day of week	0-7 (or names, 0 or 7 are sunday)

### Using multiples values

---

You may use multiples values separated by comma:

```
var cron = require('node-cron');

cron.schedule('1,2,4,5 * * * *', () => {
  console.log('running every minute 1, 2, 4 and 5');
});
```

### Using ranges

---

You may also define a range of values:

```
var cron = require('node-cron');

cron.schedule('1-5 * * * *', () => {
  console.log('running every minute to 1 from 5');
});
```

### Using step values

---

Step values can be used in conjunction with ranges, following a range with '/' and a number. e.g: **1-10/2** that is the same as **2, 4, 6, 8, 10**. Steps are also permitted after an asterisk, so if you want to say “every two minutes”, just use **\* /2**.

```
var cron = require('node-cron');

cron.schedule('* /2 * * * *', () => {
  console.log('running a task every two minutes');
});
```

### Using names

---

For month and week day you also may use names or short names. e.g:

```
var cron = require('node-cron');

cron.schedule('* * * January, September Sunday', () => {
  console.log('running on Sundays of January and September');
});
```

Or with short names:

```
var cron = require('node-cron');

cron.schedule('* * * Jan,Sep Sun', () => {
  console.log('running on Sundays of January and September');
});
```

## Cron methods

---

### Schedule

---

Schedules given task to be executed whenever the cron expression ticks.

Arguments:

- **expression** *string*: Cron expression
- **function** *Function*: Task to be executed
- **options** *Object*: Optional configuration for job scheduling.

### Options

---

- **scheduled**: A *boolean* to set if the created task is scheduled. Default *true*;
- **timezone**: The timezone that is used for job scheduling. See [IANA time zone database](#) for valid values, such as *Asia/Shanghai*, *Asia/Kolkata*, *America/Sao\_Paulo*.

**Example:**

```
var cron = require('node-cron');

cron.schedule('0 1 * * *', () => {
  console.log('Running a job at 01:00 at America/Sao_Paulo timezone');
}, {
  scheduled: true,
  timezone: "America/Sao_Paulo"
});
```

## ScheduledTask methods

---

### Start

---

Starts the scheduled task.

```
var cron = require('node-cron');

var task = cron.schedule('* * * * *', () => {
  console.log('stopped task');
}, {
  scheduled: false
});

task.start();
```

## **Stop**

---

The task won't be executed unless re-started.

```
var cron = require('node-cron');

var task = cron.schedule('* * * * *', () => {
  console.log('will execute every minute until stopped');
});

task.stop();
```

## **Validate**

---

Validate that the given string is a valid cron expression.

```
var cron = require('node-cron');

var valid = cron.validate('59 * * * *');
var invalid = cron.validate('60 * * * *');
```