

How to Run Cron Jobs in Node.js ?

Last Updated : 17 Oct, 2024



Cron jobs are scheduled tasks that run at specific intervals in the background, commonly used for maintenance or repetitive tasks. Users can schedule commands the OS will run these commands automatically according to the given time. It is usually used for system admin jobs such as **backups, logging, sending newsletters, subscription emails and more.**

Prerequisites:

- [Node.js](#) installed
- [NPM](#) installed

To run the cron-jobs in Node.js we will be using the node-cron library. Install and import it into your project. You then define a cron expression that determines the task's schedule and use `cron.schedule()` to execute functions at specified intervals.

Syntax:

```
cron.schedule("* * * * *",
function() {
    // Task to do
});
```

Steps to Create Node App

Step 1: Go to the project directory and run this command in the terminal

```
npm init -y
```

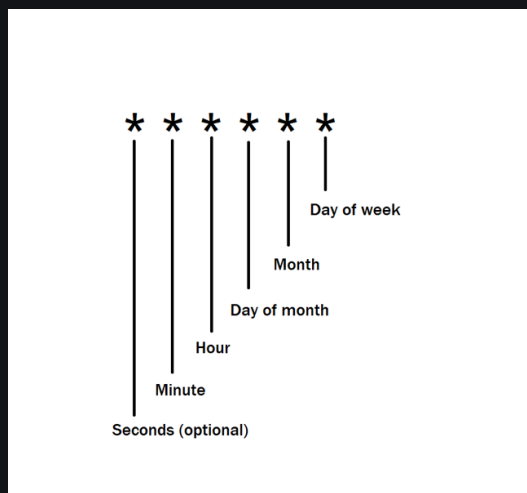
Step 2: Install the required Modules

```
npm install express node-cron
```

The updated dependencies in the package.json are:

```
"dependencies": {  
  "express": "^4.18.2",  
  "node-cron": "^3.0.0"  
}
```

Time formatting for Cron jobs



Descriptors with their ranges

- **Seconds (optional):** 0 – 59
- **Minute:** 0 – 59
- **Hour:** 0 – 23
- **Day of the Month:** 1 – 31
- **Month:** 1 – 12
- **Day of the week:** 0 – 7 (0 and 7 both represent Sunday)

Examples:

- `(*/10 * * * *)` – Runs on every 10 minutes
- `(* * 21 * *)` – Runs 21th of every month
- `(0 8 * * 1)` – Runs 8 AM on every Monday

Example: Create a new file named `index.js` and add the following code:

JavaScript

```
1 // Importing required
2 const cron = require("
3 const express = requir
4
5 app = express(); // Ir
6
7 // Creating a cron job
  second
8 cron.schedule("*/10 *
9     console.log("runni
10 });
11
12 app.listen(3000);
```

Run the file using command

```
node index
```

, you will see the output like below:

```
Lenovo@DESKTOP-6NPSGPK MINGW64 /d/Projects/random/node-cron-tut
$ node index
running a task every 10 second
running a task every 10 second
█
```

Writing to a log file

ron jobs can be used to schedule **logging tasks** in a system. We can log server status for a given time for monitoring purposes.

Example: This example sets up a cron job using node-cron to log a status message to logs.txt every 10 seconds.

JavaScript

```
1 // index.js
2 // Importing required
3 const cron = require("node-cron");
4 const express = require("express");
5 const fs = require("fs");
6
7 app = express();
8
9 // Setting a cron job
10 cron.schedule("*/10 * * * *", () => {
11
12     // Data to write to file
13     let data = `${new Date().toLocaleString()}: Server is running\n`;
14
15     // Appending data to file
16     fs.appendFile("logs.txt", data, (err) => {
17         if (err) throw err;
18         console.log("Status message logged successfully");
19     });
20 });
21
22 app.listen(3000);
```

After running above file for 30-40 seconds, you will see a file created named logs.txt having content somewhat similar to the following:

```
JS index.js logs.txt X
logs.txt
1 Sat, 18 Jan 2020 11:36:30 GMT : Server is working
2 Sat, 18 Jan 2020 11:36:40 GMT : Server is working
3 Sat, 18 Jan 2020 11:36:50 GMT : Server is working
4 Sat, 18 Jan 2020 11:37:00 GMT : Server is working
5
```

Monthly Newsletters

Sending monthly newsletters are also a use case for cron jobs in which an email will be sent to the users monthly with the latest products or blog information on the website. You can learn more about sending email in Node.js [here](#).

Example: This example uses node-cron to send an email every minute with Nodemailer, automatically triggering the `sendMail()` function.

JavaScript

```
1 // index.js
2 // Importing packages
3 const cron = require("node-cron");
4 const express = require("express");
5 const nodemailer = require("nodemailer");
6
7 app = express();
8
9 // Calling sendEmail()
10 cron.schedule("*/*/* * * *", () => {
11   sendMail();
12 });
13
14 // Send Mail function
15 function sendMail() {
16   let mailTransporter = nodemailer.createTransport({
17     service: "gmail",
18     auth: {
```

```

19         user: "<your-email@domain.com>"
20         pass: "*****"
21     }
22 });
23
24     // Setting credentials
25     let mailDetails = {
26         from: "<your-email@domain.com>",
27         to: "<user-email@domain.com>",
28         subject: "Test Email",
29         text: "Node.js is awesome"
30         + " testing email sending"
31     };
32
33
34     // Sending Email
35     mailTransporter.sendMail(mailDetails, function(err) {
36
37         if (err) {
38             console.log("Error in sending email: ", err);
39         } else {
40             console.log("Email sent successfully");
41         }
42     });
43 }
44
45 app.listen(3000);

```



The above script will send emails every minute.

Note: Open [this](#) link to **Allow less secure apps: ON**.

Now run the file using `node index.js`, you will see the output like below:

In console:

```

Lenovo@DESKTOP-6NPSGPK MINGW64 /d/Projects/random/node-cron-tut
$ node index
Email sent successfully
Email sent successfully

```

Share Your
Experiences

How to Run
Cron Jobs in
Node.js ?

How to Run
Java Code in
Node.js ?

How To Create
Modules in
Node.js?

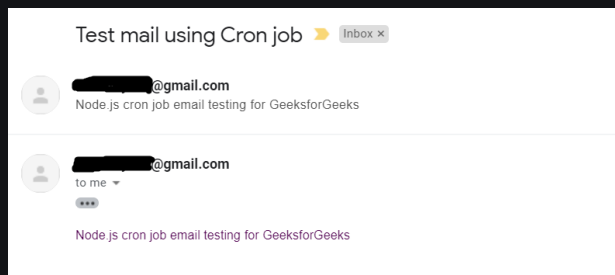
How to Install
Node & Run
npm in VS Code?

How to Copy a
File in Node.js?

How to Exit
Process in
Node.js ?

How to Take

In Gmail Inbox:



< [Previous Article](#)

[Next Article](#) >

How to work
Mongojs module
in Node.js ?

Similar Reads

Run Python script from Node.js using

Node.js is one of the most adopted web development technologies but it lacks

🕒 3 min read

How to Run Synchronous Queries using

The sync-sql module is designed to make synchronous queries to the

🕒 2 min read

Run Python Script using PythonShell

Nowadays Node.js is the most attractive technology in the field of backend

🕒 3 min read

How to Run Multiple Versions of Node.js

Usually, we work on different versions for our Node.js project and it's hard to

🕒 2 min read

How to run a node.js application

NodeJS is a runtime environment on a V8 engine for executing JavaScript code

🕒 2 min read

How to Run Node.js with Express on

Node.js is a cross-platform open-source Javascript runtime environment and

🕒 2 min read

How to Install Node & Run npm in VS

Node is an open-source, server-side JavaScript runtime environment built on

🕒 2 min read

How to Run a Node.js App as a

Running a Node.js application as a background service is a common

🕒 2 min read

How to Run Java Code in Node.js ?

Running Java code within a Node.js environment can be useful for

🕒 2 min read

How to Run Node.js Program as an

Node.js allows you to create standalone executables from your JavaScript

🕒 2 min read

Article Tags :

[Node.js](#)[Technical Scripter](#)[Web Technologies](#)[Node.js-Misc](#)[+1 More](#)

Corporate & Communications
Address:- A-143, 9th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305) | Registered Address:- K 061,

Company

About Us
Legal
Careers
In Media
Contact Us
Advertise with us
GFG

Explore

Job-A-Thon
Hiring
Challenge
Hack-A-Thon
GfG Weekly
Contest
Offline
Classes

Languages

Python
Java
C++
PHP
GoLang
SQL
R Language
Android

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap

Data Science & ML

Data Science With Python
Data Science For Beginners
Machine Learning
ML Maths

Tower K, Gulshan
Vivante Apartment,
Sector 137, Noida,
Gautam Buddh
Nagar, Uttar
Pradesh, 201305



Corporate
Solution
Placement
Training
Program

(Delhi/NCR)
DSA in
JAVA/C++
Master
System
Design
Master CP
GeeksforGeeks
Videos
Geeks
Community

Tutorial

DSA Interview
Questions
Competitive
Programming

Data
Visualisation
Pandas
NumPy
NLP
Deep
Learning

Python
Tutorial

Python
Programming
Examples
Django
Tutorial
Python
Projects
Python
Tkinter
Web Scraping
OpenCV
Tutorial
Python
Interview
Question

Computer
Science

GATE CS
Notes
Operating
Systems
Computer
Network
Database
Management
System
Software
Engineering
Digital Logic
Design
Engineering
Maths

DevOps

Git
AWS
Docker
Kubernetes
Azure
GCP
DevOps
Roadmap

System
Design

High Level
Design
Low Level
Design
UML
Diagrams
Interview
Guide
Design
Patterns
OOAD
System
Design
Bootcamp
Interview
Questions

School
Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English
Grammar

	Databases	Preparation	Competitive	More	Free Online
	SQL	Corner	Exams	Tutorials	Tools
	MYSQL	Company-	JEE Advanced	Software	Typing Te
	PostgreSQL	Wise	UGC NET	Development	Image Ed
	PL/SQL	Recruitment	UPSC	Software	Code
	MongoDB	Process	SSC CGL	Testing	Formatte
		Resume	SBI PO	Product	Code
		Templates	SBI Clerk	Management	Converte
		Aptitude	IBPS PO	Project	Currency
		Preparation	IBPS Clerk	Management	Converte
		Puzzles		Linux	Random
		Company-		Excel	Number
		Wise		All Cheat	Generato
		Preparation		Sheets	Random
		Companies		Recent	Password
		Colleges		Articles	Generato
	DSA/Placements	Development/Testing	Machine Learning/Data Science	Programming	Clouds/DevOps
	DSA - Self	JavaScript	Learning/Data	Languages	DevOps
	Paced Course	Full Course	Science	C	Engineer
	DSA in	React JS	Complete	Programming	AWS
	JavaScript -	Course	Machine	with Data	Solutions
	Self Paced	React Native	Learning &	Structures	Architect
	Course	Course	Data Science	C++	Certificat
	DSA in Python	Django Web	Program -	Programming	Salesforc
	- Self Paced	Development	[LIVE]	Course	Certified
	C	Course	Data Analytics	Java	Administ
	Programming	Complete	Training using	Programming	Course
	Course Online	Bootstrap	Excel, SQL,	Course	
	- Learn C with	Course	Python &	Python Full	
	Data	Full Stack	PowerBI -	Course	
	Structures	Development	[LIVE]		
	Complete	- [LIVE]	Data Science		
	Interview	JAVA Backend	Training		
	Preparation	Development	Program -		
	Master	- [LIVE]	[LIVE]		
	Competitive	Complete	Mastering		
	Programming	Software	Generative AI		
	Core CS	Testing	and ChatGPT		
	Subject for	Course [LIVE]			
	Interview	Android			
	Preparation	Mastery with			

Mastering
System
Design: LLD to
HLD
Tech
Interview 101
- From DSA to
System
Design [LIVE]
DSA to
Development
[HYBRID]
Placement
Preparation
Crash Course
[LIVE]

Kotlin [LIVE]