interweb

Generated on 15/11/2026 00:13

KNOWLEDGE BASE · PUBLIC DOCUMENT

Running a Node.js script on a schedule in cPanel

Cron Jobs

CODE	VERSION	COUNTRY · LANGUAGE
RLX-AO-KB-1058	v2026.10.18	Angola · English

Interweb Angola
RLX Angola, Lda. · Group RLX Holdings
Tax ID (NIF) 500 260 5724 · Commercial registration no. 26464-25/250724

https://www.meu.interweb.ao/knowledgebase/1058

Running a Node.js script on a schedule in cPanel

Cron Jobs Created on **October 10, 2026** Updated on **October 18, 2026** 3 min read



To schedule a Node.js script in cPanel, create a **Node.js application** (to get the right version and the `node_modules` folder) and make the **cron job call that environment's** `node`, with the script's full path. Just calling "node" in cron will not do: cron does not know the version you chose.

Step by step

1. **Create the application in "Setup Node.js App".** Choose the version and the application folder. It works as the environment even if your script never answers web requests. See **Node.js in cPanel: choosing the version and using npm**.

2. **Install the dependencies** with the `npm` of the application's environment, from `package.json`. The application's install button does the same.

3. **Put the script in the application folder**, or in a folder of your own outside `public_html`. Use full paths for any file it reads or writes.

4. **Find the environment's `node`.** The application screen shows the command that activates the environment, with a path ending in `bin/activate`. Copy the path and swap `activate` for `node`.

5. **Write the cron line:** the environment's `node` path, the script path, and `>> /home/ACCOUNT/job.log 2>&1`. For the schedule see **cron schedule examples**.

6. **Keep the configuration in a file of your own**, outside `public_html`. The environment variables on the application screen belong to the application, and cron does not see them.

What usually goes wrong

Symptom	Cause	What to do
`node: command not found`	Cron does not have `node` on its path.	Use the full path of the environment's `node`.
`Cannot find module`	The dependencies were not installed, or the script runs in another folder.	Install in the environment and give a full path to the script.
Wrong version	Cron calls a different version from the one the application uses.	Point at the `node` of the chosen version.
The script never ends	A connection is left open, or a timer stays active.	End explicitly with `process.exit()` when done, or close the connections.
"Out of memory" error	The script reads everything into memory.	Process in parts, or use a VPS for large jobs.

Tell cron whether it went well. End the script with a non-zero exit code when something fails (for example by setting `process.exitCode = 1`) and write the error to the output. That way cron's e-mail, or a notice of your own, only appears when there is a reason. For the notice, see **alerts to Telegram**.

Cron is for jobs that finish.
A script that stays up all the time (a server, a bot) is another matter: on a shared plan it runs as a Node.js application managed by the panel, and on a VPS it can run as a service. See **keeping a Node.js app running on a VPS with PM2**.

Test the script by hand before you schedule it
, in the cPanel Terminal if your account has one, with the same line you will put in cron. If it works there, only the schedule is left.

RECOMMENDED

Want a plan with Node.js and cron in cPanel? See the hosting plans.

See the plans →

Where to see it in your account

The Node.js versions you can use are the ones your cPanel offers, so read them there: open **Setup Node.js App** and start creating an application, and the available versions appear in the version field. If the Setup Node.js App icon is not in your cPanel, ask us whether your plan can have it, because the recipe above depends on it. To find out whether your account also has a command line, look for **Terminal** or **SSH Access** in cPanel: if either appears, the account has SSH. For a specific Node.js version or a script that has to stay running, a VPS gives you full SSH access and lets you install what you need.

SEE ALSO

- **Node.js in cPanel: choosing the version and using npm**
- **Getting a Node.js application running**
- **Cron jobs: what they are for and how to create one**

interweb

Interweb Angola

RLX Angola, Lda. · Group RLX Holdings

TAX ID (NIF)	500 260 5724
COMMERCIAL REGISTRATION NO.	26464-25/250724
REGISTERED OFFICE	Rua Direita do Patriota, Bairro Dangereux Belas, Luanda, Angola
CONTACT	info@interweb.ao · +244 942 011 191 · www.interweb.ao
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-AO-KB-1058 · v2026.10.18 · Generated on 15/11/2026 00:13