interweb

Generated on 15/11/2026 00:11

KNOWLEDGE BASE · PUBLIC DOCUMENT

Running a PHP script from a cron job

Frameworks, Composer & Git

CODE	VERSION	COUNTRY · LANGUAGE
RLX-AO-KB-774	v2026.10.10	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/774

Running a PHP script from a cron job

Frameworks, Composer & Git Created on **October 10, 2026** 2 min read



In **Cron Jobs**, in cPanel, the command is the full path of PHP, a space, and the full path of the script: `/opt/alt/php82/usr/bin/php /home/YOURACCOUNT/scripts/task.php`. Two things ruin most jobs that "do not run": **relative paths** and **the wrong PHP**. For what cron itself is, see **cron jobs**.

Setting the job up

1. **Keep the script outside the public folder**, for example in a `scripts` folder next to `public_html`. A job script inside the site can be called by anyone from a browser.
2. **Pick the PHP.** The path looks like `/opt/alt/php82/usr/bin/php`, with 82 swapped for the project's version (no dot). Do not write only `php`: cron may use a different version from your folder's.
3. In cPanel, under **Advanced**, open **Cron Jobs**. Choose the frequency and paste the full command.
4. **Send the output to a file.** Add `>> /home/YOURACCOUNT/scripts/task.log 2>&1` at the end. Without it, cPanel e-mails the output or drops it.
5. Wait for the first run and **open the log file**. If it is empty and the script should have written something, the job did not run.

A complete line looks like this:

```
/opt/alt/php82/usr/bin/php /home/YOURACCOUNT/scripts/task.php >>
/home/YOURACCOUNT/scripts/task.log 2>&1
```

What changes between the browser and cron

In the browser	In cron
There is a web request: `$_SERVER`, `$_GET` and the headers exist.	There is no request at all. Those variables are empty or missing.
The working folder is the script's.	The working folder is another one. Use `__DIR__` in `require` and file paths.
There is a time limit per request.	On the command line the time limit is usually more generous, but memory and the account's limits still apply.
The configuration is the folder's.	The configuration is the command-line PHP of that version. A value set only in the site folder's `.user.ini` may not apply here.

The clock is the server's
, time zone Africa/Maputo (UTC+02:00). A job "at 8" runs at 8 on that clock, which may not be yours. A `%` inside the command must be written `\%` or cron cuts the line: put any date logic inside the script.

Do not run a job every minute "just to see".
Each run spends processes and memory from the account (see **the limits**). Pick the frequency the work needs and time how long the script takes.

Test by hand first.
In the **Terminal** run the same line, without cron. If it fails there, it fails in cron. If the script calls WordPress, see **swapping WP-Cron for a real cron job**.

NEED A HAND?

The job exists, the log shows an error and you cannot tell which? Send us the cron line and the text of the log.

Open a support ticket →

SEE ALSO

Cron jobs: what they are for and how to create one →

- **Swapping WP-Cron for a real server cron job**
- **The cPanel Terminal**
- **Automatic backups of your own application: cron and mysqldump**

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

RLX-AO-KB-774 · v2026.10.10 · Generated on 15/11/2026 00:11

Online version of this document