interweb

Generated on 15/11/2026 00:09

KNOWLEDGE BASE · PUBLIC DOCUMENT

admin-ajax eating your resources: how to find out and cut it down

Slow Site & Server Load

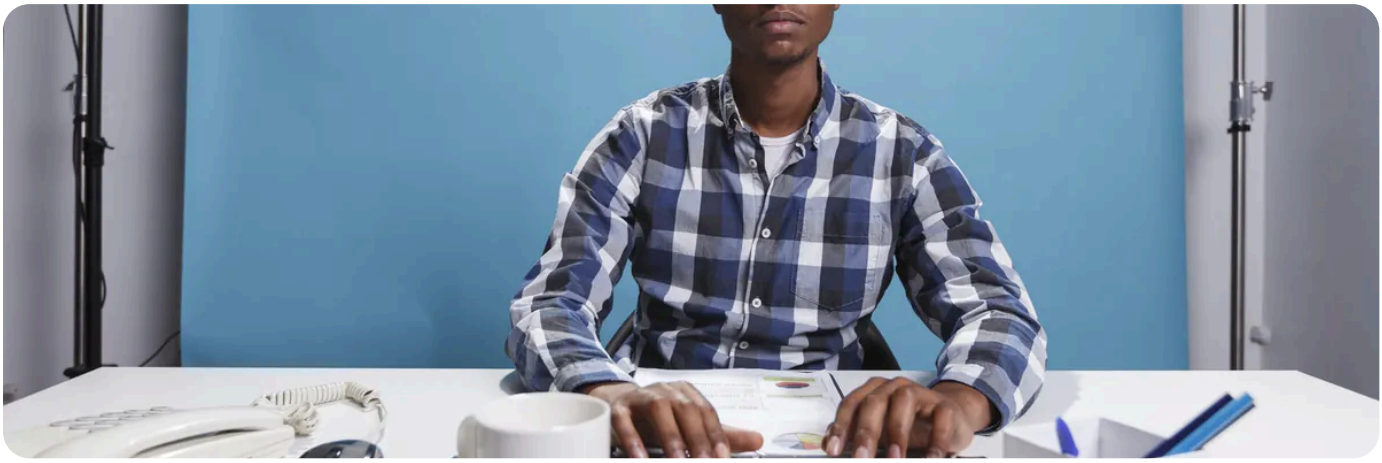
CODE	VERSION	COUNTRY · LANGUAGE
RLX-AO-KB-618	v2026.09.27	Angola · English

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/618

admin-ajax eating your resources: how to find out and cut it down

Slow Site & Server Load Created on **September 27, 2026** 4 min read



`admin-ajax.php` is the door through which *plugins* talk to the server in the background, without reloading the page. It serves the dashboard and it serves the public side of the site too: carts, filters, forms, live search. It is not a defect, and it does not get switched off.

The problem is elsewhere: **every call boots the whole of WordPress**, is never cached, and occupies a concurrent-request slot exactly like a visitor. On a site where something calls `admin-ajax.php` every five seconds, that is thousands of boots a day competing with people trying to see the site.

How to tell whether this is it

Three ways, from quickest to most complete:

1. **In the browser.** Open the site, press F12, go to the **Network** tab and type `admin-ajax` in the filter box. Leave the page open for a minute without touching anything. If the list keeps growing, you have found it.

2. **In the dashboard.** Do the same with `/wp-admin` open, especially in the page editor. A few requests there are normal; what is not normal is a constant stream.

3. **In cPanel.** Under **Metrics**, the raw access log shows every request. Count the lines containing `admin-ajax.php` in a day and compare with the number of pages served. If they are the same order of magnitude, half your server is working on this.

Cross it with the consumption.
If **Resource Usage** shows concurrent-request peaks at the same hours `admin-ajax.php` spikes, it is proven: **monitoring your site performance in cPanel** and **the limits nobody advertises**.

The usual culprits

Who is calling	What you see and what to do
The editor heartbeat	While a page is open in the editor, WordPress talks to the server every few seconds to autosave and to flag that it is being edited. One forgotten tab left open all afternoon is hundreds of requests. It is cause number one on small sites.
Cart fragments	In a shop, the cart total in the header is refreshed by a call on **every** page, including the ones with no cart at all. It is the heaviest case we see.
Statistics inside WordPress	*Plugins* that record every visit into your database make one call per visit and write one row at a time. They cost twice: the server and the database.
Live search and filters	One call per letter typed into the search box. It looks elegant and it is expensive.
Sliders and counters	Visit counters, «X people are viewing this», polls, stock notices.

What to do, in order

1. **Identify the caller.** In the **Network** tab, click one of the requests and look at the `action` field in its body. That action name usually carries the *plugin* name inside it, and then you know who to talk to.

2. **Slow the heartbeat.** There are *plugins* made only for this: they let you choose how often the editor talks to the server, and switch it off on pages where it is pointless. It is the quickest change and the lowest risk.

3. **Turn off cart fragments** on pages that are not shop pages. Most shop themes have that option; if yours does not, there are *plugins* that add it.

4. **Move statistics out of WordPress.** An outside statistics service costs your server nothing. If you would rather not use outside services, cPanel already counts your visits: **visitor statistics**.

5. **Review the live features.** Instant search, the online-visitor counter and the automatic slider rarely pay for what they cost.

Do not block `admin-ajax.php`.
It is the first thing the forums suggest and it breaks the site: forms that stop sending, carts that stop adding up, filters that do not filter, and none of it throws a visible error. The same applies to protecting the `wp-admin` folder with a password, which catches it on the way: **protecting the WordPress login**.

The cousin people confuse with this: `wp-cron.php`

If the access log shows a lot of `wp-cron.php` rather than `admin-ajax.php`, the problem is similar but the cure is different: the WordPress internal scheduler runs on the back of visits. It is dealt with in **swapping WP-Cron for a real server cron job**.

And if what you see is a lot of `/wp-login.php` or `xmlrpc.php`, that is not your site working: those are sign-in attempts from outside. They are dealt with in **protecting the WordPress login**.

What this fixes, and what it does not

Cutting background calls does not make pages faster to build: it leaves the server freer to build them. The gain shows at peak hours, in errors that used to clear by themselves, and in a dashboard that stops dragging. For the rest, the list in order of payback is in **the site is slow: what actually makes a difference**, and caching for the public pages in **installing and tuning a page cache**.

NEED A HAND?

Want us to go through the access log with you? Send the domain and the time of the peak.

Open a request →

SEE ALSO

- WordPress hosting →
- Hosting plans and the limits of each →
- Support Policy →

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-618 · v2026.09.27 · Generated on 15/11/2026 00:09