Mozout

Generated on 27/10/2026 23:05

KNOWLEDGE BASE · PUBLIC DOCUMENT

Where the PHP error log is, how to read it, and the errors you will see most

PHP & Development

CODE	VERSION	COUNTRY · LANGUAGE
RLX-MZ-KB-351	v2026.10.20	Mozambique · English

MozOut Mozambique
RLX Holdings, S.A. · Group RLX Holdings
Tax ID (NUIT) 401 191 607 · Legal entity no. (NUEL) 105 045 974 · Economic activity code (CAE) 62022

https://www.meu.mozout.co.mz/knowledgebase/351

Where the PHP error log is, how to read it, and the errors you will see most

PHP & Development Created on **September 11, 2026** Updated on **October 20, 2026** 6 min read



When a site breaks there are two ways to react: guess, or **read what the server already wrote down**. The second is faster and does not require knowing how to program - only knowing where to look. And the good news is that **logging is already on**: there is nothing to enable.

How it is configured, by default

Setting	How it is, and why
log_errors	**On.** Everything PHP refuses is written to a file, from day one.
display_errors	**Off.** Errors do *not* appear on the page - no visitor sees file paths or database names.
error_reporting	Records everything except deprecation notices, which on a site with a few years behind it would be thousands of lines a day.

It is the right combination for a live site: record everything, show nothing. What many people do - switch error display on to see what is happening and forget to switch it off - is precisely what should not be done.

Where the file is - and why you cannot find it

There is no *one* error log. There is **one per folder**: PHP writes a file called `error_log` **in the folder where the failing script lived**. That is why so many people look in a central place and find nothing.

```
public_html/error_log
```

1. In **cPanel**, open **File Manager** and turn on *Settings → Show Hidden Files*.
2. Start with `public_html/error_log`. It is the most likely one.

3 If the failing page is in a subfolder, **look in that subfolder** - its file may hold lines the root one does not.

4 Right-click and choose *Edit* or *View*. In *Terminal*, `tail -50 public_html/error_log` shows the last 50 lines, which is nearly always what matters.

Do not confuse it with the `logs` folder.
Your account's `~/logs/` holds something else: the **access** log, month by month and compressed - who visited, when, and which page. Useful for understanding traffic, useless for understanding a PHP error. For traffic and consumption, see **how to monitor performance in cPanel**.

Reading one line

They all have the same shape, and only three pieces matter:

```
[10-Sep-2026 15:02:38 UTC] PHP Warning:  include(): Failed opening
'includes/footer.php' for inclusion in /home/account/public_html/index.php on line 88
```

The piece	What it tells you
The date, in brackets	If it is not from just now, it is not your problem. Compare with the time the site failed - see **what time the server is on**
The type	**Fatal error** stopped the page. **Warning** and **Notice** stopped nothing - they are noise, until there are many.
The file and line, at the end	**This is the part that solves it.** It says exactly where to look. If the path contains *wp-content/plugins/*, you already know whose fault it is.

A huge log is itself a symptom.
We have seen `error_log` files running to **megabytes**, thousands of identical lines repeated on every visit. None of them breaks the site, so nobody notices - but each one is work the server does for nothing, and the file eats your account's space. If you find the same warning thousands of times, it is worth fixing: it is usually an out-of-date plugin or a wrong file path.

The errors you will see most, and what they mean

The message	The translation
Fatal error: Cannot redeclare fn()	The same function was declared twice. Either it is in two files, or the same file is loaded twice. The usual fix is swapping *include* for *include_once*.

The message	The translation
Fatal error: Allowed memory size exhausted	You hit the memory limit - see **PHP limits**
Fatal error: Maximum execution time exceeded	The operation took longer than allowed. Same article.
Warning: include(): Failed opening	A file the code expects is not where it looks. Notice whether the path starts with *../*: relative paths break when the page is called from another level.
Parse error: syntax error, unexpected…	A missing semicolon, brace or quote. The line number is reliable; the mistake is usually on that line or the one before.
Error establishing a database connection	Not PHP: it is the credentials in *wp-config.php* or the database itself. See **website errors: 500, 403, 404 and 508**

Turning on-screen messages on - and off again

Sometimes the log is not enough and you need to see the error on the page. You can, and it is yours to do - it is in the PHP Selector in **cPanel** - but read the warning first.

> **Never on a live site, and never left on.**
> With `display_errors` on, any visitor starts seeing **full file paths, database names and sometimes fragments of queries**. It is a map handed to anyone wanting to attack the site, and automated scanners collect exactly that. If you really must switch it on, do it for minutes, reproduce the error, and **switch it off immediately**. On WordPress there is a better route: see **common WordPress errors**.

The log is empty or missing

That means something very specific: **there were no PHP errors**. It does not mean there was no problem - it means the problem was not PHP's.

If the log is empty but the site fails	Then look at
A 500 error with nothing in error_log	Nearly always *.htaccess*. Rename it and test - **error 403 and .htaccess rules**
A 508 error, or slowness at certain hours	Resources, not code - **the site is slow**
A blank page without a single line	The error may have happened before PHP started. Confirm the chosen version in **viewing and adjusting your PHP configuration**

If the log is empty but the site fails	Then look at
It fails only for some visitors	Not the code: an address block or caching. See **website errors**

When you write to us, send the line

It is the most useful request we receive. **One line from** `error_log`, with its date and file path, is worth more than three paragraphs describing the symptom - and it usually cuts the answer from hours to minutes. Add the address of the failing page and the time you saw it fail.

Before touching code files, make sure you have a copy - **how to restore your data with JetBackup**. For what we handle and what belongs to the application, **how far our support goes**. And if the log shows files you do not recognise, that is a different matter: **how to tell if your site has been compromised**.

NEED A HAND?

Got a line from the log and not sure what it means? Send it over.

Open a ticket →

Two more errors, and the Errors page

- **Call to undefined function**, for example `mysqli_connect()` or `imagecreatefromjpeg()`: the function belongs to a PHP extension that is not switched on, or to a file the code did not load. In cPanel, **Select PHP Version**, on the **Extensions** tab, tick the one the application asks for (mysqli, gd, intl...) and reload the page.
- **failed to open stream: Permission denied**: PHP cannot read or write that file or folder. In File Manager, **Permissions**: 644 for files and 755 for folders is the usual; never 777.
- **The Errors page**: in cPanel, in the **Metrics** section, **Errors** shows the most recent entries of the web server's log for your account. It is where a 500 with an empty error_log usually shows its cause, such as a wrong line in .htaccess.

Mozout

MozOut Mozambique

RLX Holdings, S.A. · Group RLX Holdings

TAX ID (NUIT)	401 191 607
LEGAL ENTITY NO. (NUEL)	105 045 974
ECONOMIC ACTIVITY CODE (CAE)	62022
REGISTERED OFFICE	Bairro 7 de Setembro Vilankulo 1304 · CEP 0411-02, Inhambane, Mozambique
CONTACT	info@mozout.co.mz · +258 84 635 1111 · +258 86 635 1111 · www.mozout.co.mz
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-MZ-KB-351 · v2026.10.20 · Generated on 27/10/2026 23:05