interweb

Generated on 15/11/2026 00:08

KNOWLEDGE BASE · PUBLIC DOCUMENT

Getting a Laravel application running

Frameworks, Composer & Git

CODE	VERSION	COUNTRY · LANGUAGE
RLX-AO-KB-482	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/482

Getting a Laravel application running

Frameworks, Composer & Git Created on **September 27, 2026** 4 min read



Laravel is not a CMS: it is the foundation you write a PHP application on top of. So this article has two halves. The first is for anyone who wants a fresh skeleton to start from; the second is for anyone who already has an application written and wants it here.

The cPanel installer carries two Laravel series, and both want **PHP 8.2** or newer. Neither needs extra extensions: everything Laravel asks for is already on for a new account.

A. Starting from scratch, with the installer

1. In cPanel, check in **Select PHP Version** that you are on 8.2 or above: **changing your PHP version without breaking the site**.
2. Open the **Softaculous Apps Installer** and search for *Laravel*.
3. Pick the domain, the **https** protocol and the directory.
4. Install. What you get is an **empty Laravel project**, with the database created and the environment file written. From there the code is yours.

The installer is the quick way to get the boring part right:
The database, the application key and the permissions on the writable directories. Even if you intend to replace all the code, starting this way saves time.

B. Bringing your own application

The difficulty is always the same one: **Laravel serves the site from the *public* directory**, not from the project root. The directories holding configuration, code and keys sit one level above, and must never be reachable from a browser.

1. **Upload the project** by FTP or File Manager: **FTP accounts and FileZilla**. Do not upload the dependency directory or the compiled asset directory.

2. **Put the project OUTSIDE the public directory** and point the domain document root at the project *public* folder. For an addon domain or a subdomain, you choose that when creating it: **adding another domain or a subdomain**.

3. **Create the database** and its user in cPanel: **creating databases and using phpMyAdmin**.

4. **Write the environment file** with the database details, the site address and the application key. Without a key, Laravel refuses to boot.

5. **Install dependencies and run the migrations** from the command line. You need **SSH access**: some accounts have a shell and some do not.

6. **Check the writable directories really are writable.** If the site throws a 500 straight away, it is nearly always this or a missing key. See **website errors explained**.

! **On the command line, the default *php* may not be the version you chose for the site.**
That produces a baffling result: the site works in the browser while the command line tools refuse to run, or the other way round. Always use the **full path** to the PHP version you selected, both when installing dependencies and in scheduled tasks.

The scheduler and the queues

If your application uses the Laravel scheduler, it needs **one scheduled task running every minute** that calls the scheduler. That is how Laravel was designed: a single cron entry, and everything else decided inside the code. See **cron jobs**, and do not forget the full PHP path.

! **Queue workers that stay resident are not for shared hosting.**
A process that runs without stopping hits the account process ceiling and gets cut off. If your application needs that, its home is a server of its own: **VPS, Cloud or Dedicated**. On the ceilings, see **the limits of your account**.

Before it goes live

1. **Turn debug mode off.** Left on, any error shows the visitor your code, your paths and sometimes your database credentials.

2. **Check the environment file is not reachable from a browser.** If the project is placed correctly, it is not.

3. **Switch on the PHP accelerator. opcache** arrives off on a new account and is enabled in **Select PHP Version**.

4. **Take the first backup**, code and database: **making and testing your own backup**.

For what every installation has in common, see **installing a ready-made platform**. If your application is in another language, see **Django** or **Ruby on Rails**.

NEED A HAND?

Application uploaded and all you get is a 500? Send us the domain and the directory and we will read the error log with you.

Open a support ticket →

SEE ALSO

- Hosting plans and what each one includes →
- VPS Server →
- 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-482 · v2026.09.27 · Generated on 15/11/2026 00:08