interweb

Generated on 21/10/2026 17:22

KNOWLEDGE BASE · PUBLIC DOCUMENT

Getting a Node.js application running

Node.js, Python & Docker

CODE	VERSION	COUNTRY · LANGUAGE
RLX-MZ-KB-698	v2026.10.18	Mozambique · English

Interweb 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.interweb.co.mz/knowledgebase/698

Getting a Node.js application running

Node.js, Python & Docker Created on **September 27, 2026** Updated on **October 18, 2026** 5 min read



Yes, there is Node.js on this hosting. It is not an extra you ask for, and it is not a VPS thing: it is in your cPanel, under **Software**, called **Setup Node.js App**. The selector offers **thirteen versions**, all of them live, and you pick one per application.

Next to it are the same doors for the other two languages: **Setup Python App** and **Setup Ruby App**. They all work the same way, and everything said here applies to all three.

What the selector does, and what it does not

This is the part that saves you an afternoon. The tool does four things, and only four.

It does	It does not
Pick the Node version for this application, without touching the others in the account.	It does not install your application. The code is yours, and it reaches the server by FTP, by Git or through Terminal.
Build an isolated environment for it, holding that application's dependencies and nobody else's.	It does not fix code or guess what is missing. If the application crashes on startup, it still crashes.
Attach it to an address on your domain, and look after the web server in front of it.	It does not give you a public port of your own. The application is served through the domain, not through a port number.
Start, stop and restart it, and hold its environment variables.	It does not make you administrator of the machine, give you system services, or let you install packages outside the account.

Not every application fits shared hosting.

An ordinary application answering web requests runs fine. One that needs to open ports of its own, listen outside the domain, register system services or hold administrator rights on the machine does not fit here, and no tool changes that. Those cases want a VPS or a dedicated server. If you are unsure, tell us what the application needs before you buy.

The order of the steps

1. **Get the code onto the server first**, into a folder **outside** the public website folder. By FTP (**uploading with FileZilla**), from a repository (**Git in cPanel**) or over the command line (**Terminal**).

2. **Open Setup Node.js App and create the application.** You will give it the Node version, the folder holding the code, the address you want it to answer on, and the file that starts it.

3. **Install the dependencies.** The tool can do it from the file that lists them. If you would rather do it by hand, you first have to step into the application's environment: the screen itself shows you the command for that, and that is the command you paste into Terminal. Installing dependencies outside the environment puts them in the wrong place, and then the application says it cannot find them.

4. **Put keys and passwords into the environment variables**, on the application's own screen, rather than into the code. That is what they are there for.

5. **Start it, and visit the address.** If nothing answers, the application screen is the first place to look and the account error log is the second: **where the error log is**.

The right version is the one your application asks for, not the highest in the list.

The screen shows which version each application is on, and changing it later is one box. Jumping several versions at once is the fastest way to break dependencies that worked: move one step at a time and reinstall dependencies after each.

What usually goes wrong

What you see	What it is
It starts and dies straight away	Nearly always the wrong startup file, or a missing dependency. Install dependencies inside the environment and check the startup file name.
The address shows the normal site instead of the application	The address you gave the application is not the one you are visiting, or a file in that path is winning. Check both on the same screen.

What you see	What it is
The application wants a port and you have none	That is by design. Here the application is served through the domain, not through a port you open. If it only knows how to run on a port of its own, it needs a server of your own.
It stops by itself after a while	Account limits cut off whatever goes past them. See **the limits nobody advertises**.
It connects to the database on your computer but not here	Inside the server the database is local, and the address changes. **where each application keeps its connection details** covers it.

And the other languages?

Language	What the panel gives you
Python	**Setup Python App**, with a version selector from 2.7 to 3.13. For Django specifically there is a separate article: **getting a Django application running**.
Ruby	**Setup Ruby App**, from 2.2 to 3.4, and also with its own article: **getting a Ruby on Rails application running**.
PHP	Needs none of this: it runs directly, and the version is chosen per folder in **Select PHP Version**. For Laravel, **getting a Laravel application running**.

The full list of what this account can run is in **which technologies run here**.

NEED A HAND?

Got an application and want to know whether it fits here or needs its own server? Tell us what it does.

Open a support ticket →

Where to see it in your account

A Node.js application runs under the same account ceilings as the rest of your hosting. We do not publish per-plan figures for processes, memory or files, so read them in your account: Resource Usage in cPanel shows entry processes, memory and CPU against the limit, and when you reached it; the Statistics sidebar shows the file count and the space used.

Keep an eye on the file count: the dependencies folder of a Node project (node_modules) can hold tens of thousands of files. If the application needs more than a shared account gives, a VPS (full SSH access, from 1 vCPU and 2 GB RAM) is the next step.

SEE ALSO

- **VPS Server: when the machine has to be yours**
- **Hosting plans and what each one includes**
- **Support Policy: how far our help goes**

interweb

Interweb 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@interweb.co.mz · +258 84 635 1111 · +258 86 635 1111 · www.interweb.co.mz
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-MZ-KB-698 · v2026.10.18 · Generated on 21/10/2026 17:22