interweb

Generated on 15/11/2026 00:09

KNOWLEDGE BASE · PUBLIC DOCUMENT

Getting a Ruby on Rails application running

Python & Other Languages

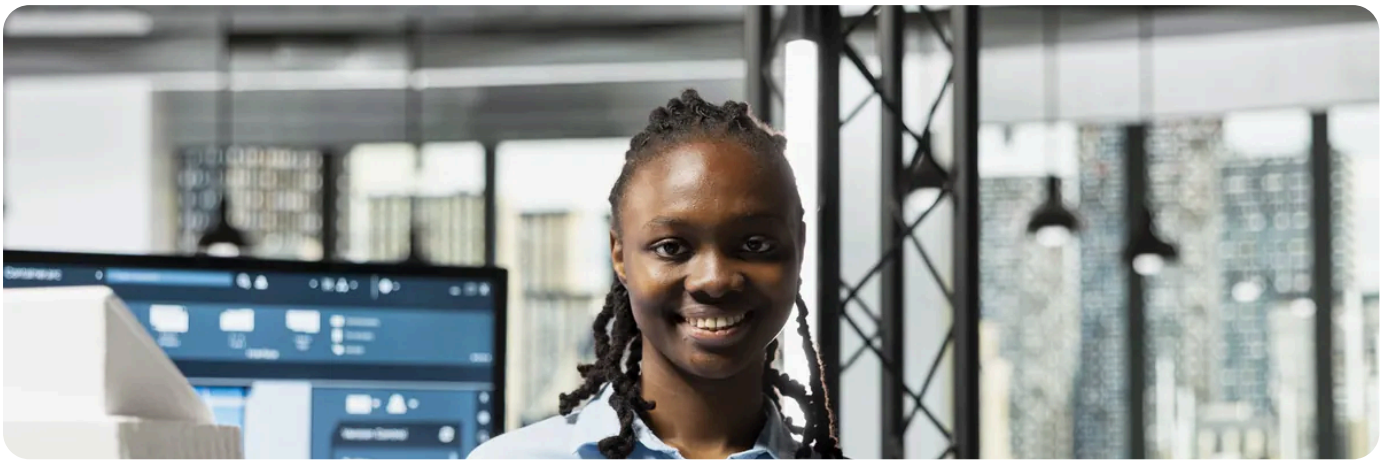
CODE	VERSION	COUNTRY · LANGUAGE
RLX-AO-KB-488	v2026.10.21	Angola · English

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/488

Getting a Ruby on Rails application running

Python & Other Languages Created on **September 27, 2026** Updated on **October 21, 2026** 4 min read



Ruby on Rails is not in the cPanel apps installer, because that installer is for PHP applications. Ruby has a tool of its own, under **Software** in cPanel: **Setup Ruby App**.

That page creates a Ruby environment of your own, picks the interpreter version, and attaches the application to an address on your domain. On the day this article was written the versions on offer ran from **2.2** to **3.4**.

> **One thing to know before you start: the *rails* command is not installed on the server.**
> That is not a fault, it is how Ruby works: Rails is a library, and it installs **inside your application environment**, not system wide. Until you install it there, typing *rails* on the command line does nothing, and that catches everyone the first time.

Creating the environment, step by step

1. Sign in to **My Interweb** and open the cPanel for the plan.
2. Under **Software**, open **Setup Ruby App** and create an application.
3. **Pick the Ruby version.** For a new project, take one of the newest in the list: that is where libraries are still being published.
4. **Give it the code directory**, which belongs outside the public web directory, and **the address** where the application will answer.
5. Create it. The page then shows you **the command that activates this environment** on the command line. Keep it.

From there, the application is served by **Passenger**, which bridges the web server and Ruby. Passenger looks for a startup file at the project root, *config.ru*, which every Rails project already ships with.

Installing Rails and the gems

1. Check the account has **SSH access**. Some accounts have a shell and some do not: without one, this route is closed and it is best to **open a support ticket**" style="color:#d60037;text-decoration:none;font-weight:bold;">open a support ticket.
2. Connect over SSH and **run the activation command first**, the one the selector page gave you. This step is the whole difference: without it you are using the system Ruby and not your application one.
3. **Install Rails as a gem** inside that environment. The *rails* command then exists, but only while the environment is active.
4. **Upload the project** into the directory you named, or generate a new one there, and install the gems from the project dependency file.
5. **Create the database** in cPanel and point the Rails configuration at it: **creating databases and using phpMyAdmin**. Then run the migrations.
6. **Restart the application** from the selector page. Without a restart, Passenger keeps serving what it had in memory.

Open a new SSH session and you have to activate the environment again.
Half the trouble people hit when starting out comes down to this: a gem installed in a session where they forgot to activate, and an application that keeps saying it cannot find it.

Before it goes live

Topic	What to do
Production environment	Put the application into production mode. In development, Rails reloads code on every request and shows errors to the visitor with your source around them.
Static assets	Precompile the assets before going live. In production Rails expects to find them ready, and a page with no styling at all is almost always this step forgotten.
Secrets	The key that decrypts the Rails credentials does not belong in the repository. Put it in the application environment variables, on the selector page.

Topic	What to do
Periodic jobs	Use cPanel scheduled tasks, calling the command inside the activated environment: **cron jobs**.
Logs	Rails logs grow without end and fill the account quietly. Rotate them on a schedule, or you meet them when the disk fills.

Queue workers and resident processes do not fit a shared plan.
The web application starts on demand and lives happily; a background job process running without stop hits the account process ceiling and gets cut off, with no warning you will see. 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**.

For the wider picture, see **which technologies run here**. If your application is in another language, see **Django** or **Laravel**.

NEED A HAND?

Environment created and the application will not start? Send us the domain and the directory and we will read the Passenger 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-488 · v2026.10.21 · Generated on 15/11/2026 00:09