Mozout

Generated on 15/11/2026 00:38

KNOWLEDGE BASE · PUBLIC DOCUMENT

Environment variables and secrets for your application

Deploy, Logs & Access

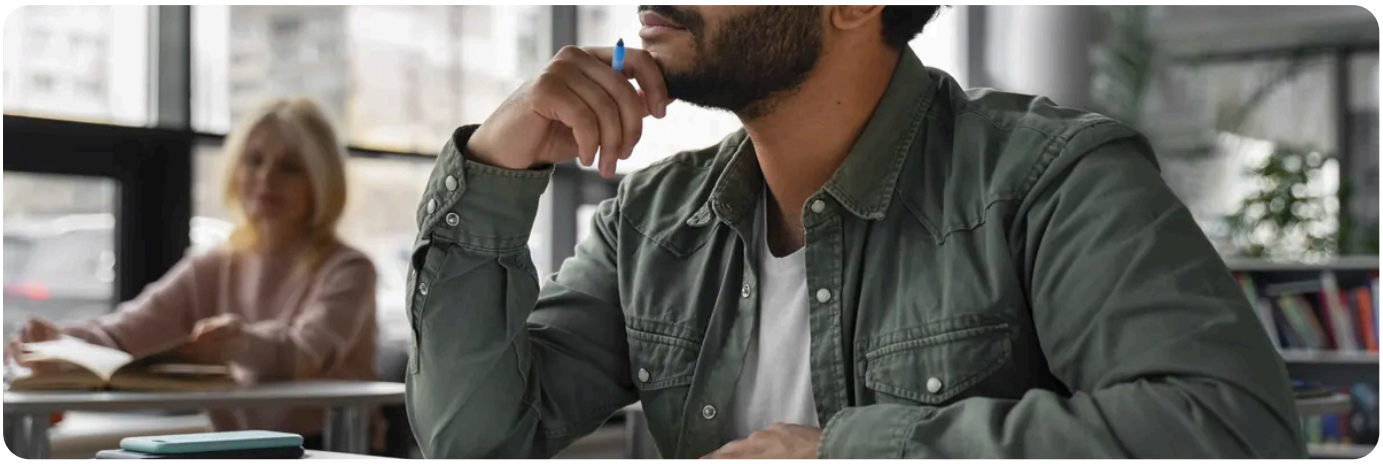
CODE	VERSION	COUNTRY · LANGUAGE
RLX-INT-KB-2790	v2026.10.10	International · English

MozOut International
RLX Holdings (Pty) Ltd. · Group RLX Holdings
Tax no. (SARS) 943 907 6234 · Registration no. (CIPC) 2025/667953/07

https://www.my.mozout.com/knowledgebase/2790

Environment variables and secrets for your application

Deploy, Logs & Access Created on **October 10, 2026** 3 min read



An **environment variable** is a value the application reads from outside the code: the database password, an API key, the address of a service. You keep it out of the code for two reasons: code goes into Git and into other people's hands, and the same code runs in different places (your machine, a test site, production) with different values. What changes is **where** you set each variable, depending on where the application runs.

Where to set it, in each case

Where it runs	Where to put the variables	Afterwards
Node.js in cPanel	On the application screen, in the environment variables part (name and value).	Save and **restart** the application.
Python in cPanel	On the application screen, in the environment variables part.	Save and **restart**.
PHP	In a `.env` file **outside** the public folder, read by the code. See **keeping passwords out of PHP code**.	The code reads it on every request.
VPS with Node.js and PM2	In `ecosystem.config.js`, or in the environment of the user running the application.	`pm2 restart name --update-env`.
VPS with systemd	In a file pointed to by `EnvironmentFile=` in the service unit.	`sudo systemctl restart name`.
Docker Compose	In a `.env` next to `compose.yaml` (with `env_file:`), or under `environment:`.	`docker compose up -d` recreates the container.

Step by step, in cPanel

1. **Write the code to read the variable,** not the value. In Node.js: `process.env.DB_PASSWORD`. In Python: `os.environ["DB_PASSWORD"]` (or `os.environ.get("DB_PASSWORD")` if it may be absent). In PHP: `getenv("DB_PASSWORD")`.

2. **Open “Setup Node.js App” or “Setup Python App”** and your application.

3. **Add the variable:** the name (by convention upper case, with underscores) and the value. Save.

4. **Restart the application.** A process only reads its environment when it starts; a new or changed variable does not reach a process that was already running.

5. **Check without showing it.** Make the application report only whether the variable exists (“set” or “missing”), never the value.

In a systemd unit, on a VPS, it looks like this:

```
[Service]
User=appuser
WorkingDirectory=/home/appuser/app
EnvironmentFile=/home/appuser/app/.env
ExecStart=/usr/bin/node app.js
Restart=on-failure
```

The file has one `NAME=value` line per variable, without `export`. Confirm the path of `node` with `which node`. Then: `sudo systemctl daemon-reload` and `sudo systemctl enable --now name`.

A secret in the code is already a lost secret.
If a password ended up in a repository, even if you delete the line afterwards, it stays in the history: **change it**. Add `.env` to `.gitignore`, give it permissions for the owner only (`chmod 600 .env`), and do not keep it in the site’s public folder. Do not write the value into logs or error messages.

There are no secrets in the browser.
Variables that front-end tools bake into the browser JavaScript (the ones with prefixes such as `NEXT_PUBLIC_` or `VITE_`) are visible to any visitor. Put only what may be public there; real keys stay on the server.

Use different values in each place

(test and production) and one key per service, with minimal permissions. If a key leaks, you change just one. For getting and storing AI keys, see **API keys for AI models**. The database connection has its own details in **connecting from Python and Node.js**.

NEED A HAND?

Set the variable and the application still does not see it? Tell us the domain and the variable name (never the value), and we will see what the server passes to the application.

Open a support ticket →

SEE ALSO

- **Keeping passwords out of your PHP code: .env files and the right permissions** →
- **Node.js in cPanel: choosing the version and using npm** →
- **Python in cPanel: virtual environments and pip** →

Mozout

MozOut International

RLX Holdings (Pty) Ltd. · Group RLX Holdings

TAX NO. (SARS)	943 907 6234
REGISTRATION NO. (CIPC)	2025/667953/07
REGISTERED OFFICE	1111 Olive Street Saint Louis 63101, Missouri, United States
CONTACT	info@mozout.com · +27 78 823 3325 · www.mozout.com
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

RLX-INT-KB-2790 · v2026.10.10 · Generated on 15/11/2026 00:38

Online version of this document