interweb

Generated on 15/11/2026 01:42

KNOWLEDGE BASE · PUBLIC DOCUMENT

# Running a Flask or FastAPI application

Python & Other Languages

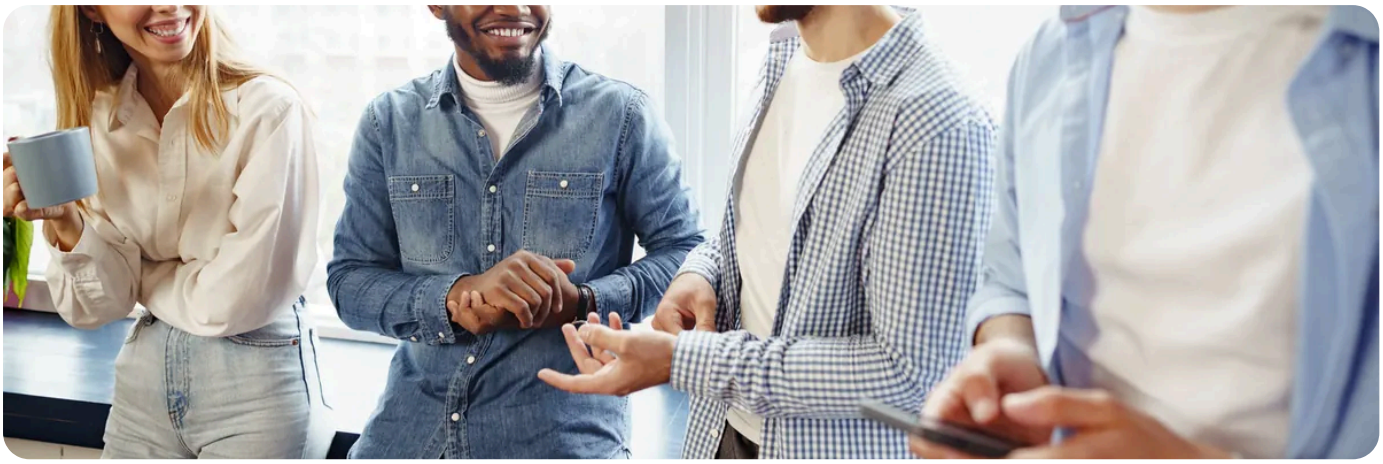
| CODE | VERSION | COUNTRY · LANGUAGE |
| --- | --- | --- |
| RLX-MZ-KB-900 | v2026.10.10 | Mozambique · English |

**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/900

# Running a Flask or FastAPI application

Python & Other Languages  Created on **October 10, 2026**  3 min read



**Flask** runs on shared hosting with no tricks: it speaks WSGI, and that is what the Passenger behind "Setup Python App" expects. **FastAPI** is different: it speaks ASGI, and Passenger does not speak that directly. You can make it work with an adapter, but you lose what makes it interesting (long connections, *websockets*). To make real use of it, the place is a **VPS**. This article shows both paths. Creating the application and the virtual environment is covered in **Python in cPanel: virtual environments and pip**.

## Which one for where

| | On shared hosting (cPanel) | On a VPS |
|---|---|---|
| **Flask** | Yes, through "Setup Python App" (WSGI). | Yes, with gunicorn behind a proxy. |
| **FastAPI** | Only with an ASGI-to-WSGI adapter. No *websockets* and no streamed responses. | Yes, with uvicorn behind a proxy: it is the intended design. |
| **Always-on processes (queues, tasks)** | No: the application only lives while there are requests. | Yes, with systemd, Docker or another manager. |
| **Who installs and maintains** | You, in your account. | You, on the unmanaged VPS. |

## Flask in cPanel

1. **Create the application** under "Setup Python App" with a 3.x version, the root and the address you want. Leave the startup file as `passenger_wsgi.py` and the entry point as `application`.

2. **Put your code in the application root.** For example, an `app.py` with `app = Flask(__name__)`.

3. **Connect Passenger to Flask:** in `passenger_wsgi.py` write just

```
from app import app as application
```

The name `application` is what Passenger looks for.

4. **Install the libraries** with a `requirements.txt` and "Run Pip Install", and **restart**.

5. **Open the address.** If it errors, the Passenger log (the *Passenger log file* field) says what failed.

## FastAPI

**In cPanel, with an adapter.** Install `fastapi` and `a2wsgi` (an ASGI-to-WSGI adapter), and in `passenger_wsgi.py`:

```
from a2wsgi import ASGIMiddleware
from main import app

application = ASGIMiddleware(app)
```

It works for an API of simple requests and responses. Do not count on *websockets*.

**On a VPS.** Inside the project's virtual environment, install `fastapi` and `uvicorn` and start it:

```
uvicorn main:app --host 127.0.0.1 --port 8000
```

The `main:app` means "the file `main.py`, the variable `app`". Listening only on `127.0.0.1`, what shows it to the world is nginx with HTTPS: **nginx as a reverse proxy**. To keep it up after you close the session, use systemd or Docker. The same goes for Flask on a VPS, swapping uvicorn for `gunicorn -b 127.0.0.1:8000 app:app`.

**Never leave debug mode on in production.**
In Flask, `debug=True` (or `FLASK_DEBUG`) opens a console in the browser that lets anyone who reaches an error page run code on the server. And do not use the development server (`flask run`) or `uvicorn --reload` as your public server.

**Test with a minimal route first:**
One that returns "ok". If that answers, the problem is in the rest of the code or in the dependencies. Secrets and configuration go in variables, not in code: **environment variables and secrets**. When the error is an unexplained 5xx, see **error 503 and the 5xx family**.

NEED A HAND?

The application opens locally but not on the server? Tell us the domain and the last line of the Passenger log.

Open a support ticket →

SEE ALSO

- **Python in cPanel: virtual environments and pip** →
- **Getting a Django application running** →
- **Nginx as a reverse proxy in front of a container** →
- **VPS servers** →

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<br>Vilankulo 1304 · CEP 0411-02, Inhambane, Mozambique |
| CONTACT | info@interweb.co.mz · +258 84 635 1111 · +258 86 635 1111 ·<br>www.interweb.co.mz |
| OPENING HOURS | Monday to Friday - 08 AM to 04 PM<br>Saturday & Sunday - Closed |

Online version of this document

RLX-MZ-KB-900 · v2026.10.10 · Generated on 15/11/2026 01:42