Mozout

Gerado a 21/10/2026 13:27

BASE DE CONHECIMENTO · DOCUMENTO PÚBLICO

Onde vão parar o console.log e o print, e como ler os registros da sua aplicação

Node.js, Python e Docker

CÓDIGO	VERSÃO	PAÍS · IDIOMA
RLX-BR-KB-3372	v2026.10.10	Brasil · Português (Brasil)

MozOut Brasil
RLX Holdings (Pty) Ltd. · Grupo RLX Holdings
N.º fiscal (SARS) 943 907 6234 · N.º de registo (CIPC) 2025/667953/07

https://www.my.mozout.com/knowledgebase/3372

Onde vão parar o console.log e o print, e como ler os registros da sua aplicação

Node.js, Python e Docker Criado em **10 de outubro de 2026** 3 min de leitura



No seu computador, o `console.log` (Node.js) e o `print` (Python) aparecem na tela. No servidor não há tela, e a pergunta é onde foram parar. **A resposta curta:** na hospedagem compartilhada, o que a aplicação escreve vai para o **registro do Passenger**, o arquivo que indica no campo *Passenger log file* da tela da aplicação; num VPS, depende de quem a inicia (PM2, systemd ou Docker). Sem esse registro, depurar uma aplicação é adivinhar.

Onde está o registro, em cada caso

Onde corre	Onde ver	Comando ou local
Node.js ou Python no cPanel	No arquivo do campo *Passenger log file* (tela do «Setup Node.js App» ou do «Setup Python App»).	Indique um arquivo dentro da pasta da aplicação, guarde, reinicie e repita o pedido. Abra-o no Gerenciador de Arquivos, ou no Terminal com `tail -n 100 arquivo.log` .
VPS com PM2	Nos registros do PM2.	`pm2 logs nome` ou `pm2 logs --lines 100`
VPS com systemd	No diário do sistema.	`journalctl -u nome -n 100` , e `-f` para seguir ao vivo
Docker	Nos registros do contêiner.	`docker logs nome` ou `docker compose logs -f serviço`

O registro geral de erros do site é outra coisa: apanha erros do servidor web, não o que a sua aplicação escreve. Para esse, veja **erros 500, 403, 404 e 508**.

Pôr a escrever registros que se leem

1. **Defina o arquivo de registro** na tela da aplicação (no cPanel), ou use o do PM2, do systemd ou do Docker (num VPS).

2 **Escreva com sentido.** Em vez de `console.log("aqui")`, diga o quê e com que valor: o método e o caminho do pedido, o identificador do registro que foi lido, o erro completo. Nunca escreva senhas, chaves nem dados pessoais.

3 **Em Python, evite o atraso.** Quando a saída vai para um arquivo, o `print` pode ficar guardado em memória e só aparecer mais tarde. Use `print("texto", flush=True)`, ou o módulo `logging`, que escreve de imediato. A variável de ambiente `PYTHONUNBUFFERED=1` tem o mesmo efeito para toda a aplicação.

4 **Reinicie e repita.** Faça o pedido que falha e leia as últimas linhas: normalmente a causa está lá. Se a aplicação nem inicia, veja **as causas do erro 503**.

! **Um registro sem limite acaba por encher o disco.**
Numa conta compartilhada, um arquivo que cresce todos os dias gasta espaço, e o espaço é limitado. Apague ou arquive de vez em quando, ou limite o tamanho (em Python, o `RotatingFileHandler` do módulo `logging` faz isso). Num VPS, o diário do sistema e os registros do Docker também crescem: veja **o Docker a encher o disco**.

Em tempo real, no Terminal.
Se a conta tem terminal (**o Terminal do cPanel**), `tail -f arquivo.log` mostra as linhas à medida que chegam: abra o site em outra aba e veja o que aparece. Para sair, Ctrl+C.

PRECISA DE AJUDA?

Tem o registro à frente e não percebe a última linha? Envie-nos o domínio, a hora do erro e essa linha.

Abrir um pedido de suporte →

VEJA TAMBÉM

- **Aplicação Node.js no cPanel com erro 503: as causas habituais** →
- **Erros do site explicados: 500, 403, 404, 508 e o que fazer** →
- **O Terminal do cPanel: uma linha de comandos sem instalar nada** →

Mozout

MozOut Brasil

RLX Holdings (Pty) Ltd. · Grupo RLX Holdings

N.º FISCAL (SARS)	943 907 6234
N.º DE REGISTO (CIPC)	2025/667953/07
SEDE	72 York Road, Kensington Joanesburgo 2094, Gauteng, África do Sul
CONTACTOS	info@mozout.com · +27 78 823 3325 · www.mozout.com
HORÁRIO	2ª. à 6ª. Feira - 08 à 16 Horas Sábado & Domingo - Fechado

Versão online deste documento

RLX-BR-KB-3372 · v2026.10.10 · Gerado a 21/10/2026 13:27