Mozout

Gerado a 14/11/2026 22:20

BASE DE CONHECIMENTO · DOCUMENTO PÚBLICO

WebSockets e Socket.IO: o que esperar no cPanel e num VPS

Node.js e JavaScript

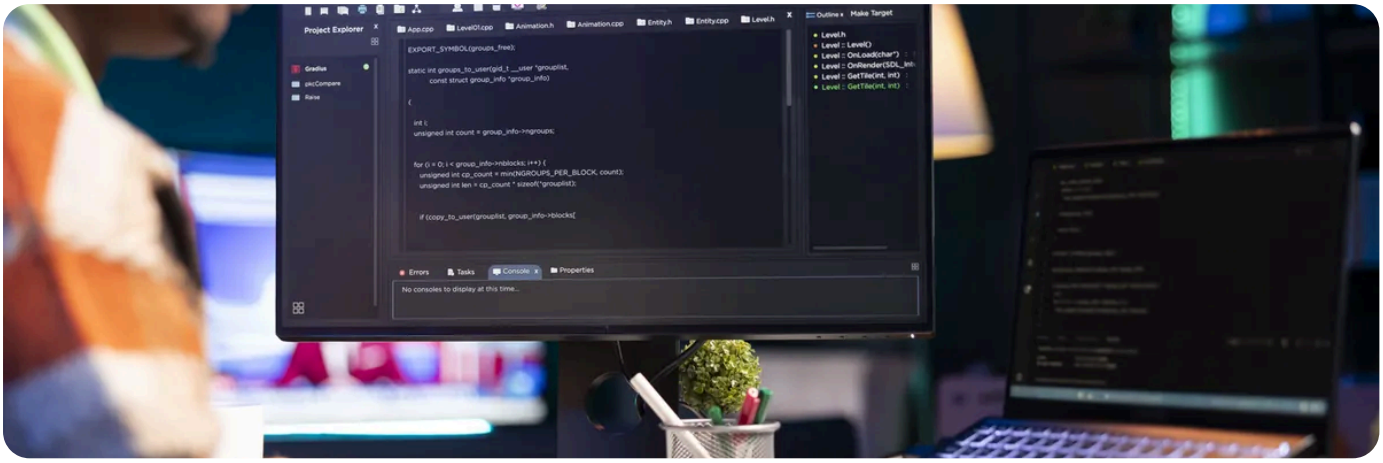
CÓDIGO	VERSÃO	PAÍS · IDIOMA
RLX-BR-KB-3402	v2026.10.18	Brasil · Português (Brasil)

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/3402

WebSockets e Socket.IO: o que esperar no cPanel e num VPS

Node.js e JavaScript Criado em **10 de outubro de 2026** Atualizado em **18 de outubro de 2026** 4 min de leitura



Um chat, um painel que se atualiza sozinho, notificações ao vivo: tudo isso pede uma conexão que fica **aberta** entre o navegador e o servidor, em vez do pedido e resposta de sempre. Essa ligação chama-se WebSocket. **A resposta curta:** numa hospedagem compartilhada, não conte com ela. Os links abertos ocupam recursos da conta durante todo o tempo em que alguém está no site, e a conta tem limites de processos. Num **VPS** funciona bem, com o proxy bem configurado.

O que muda de uma ligação normal

Uma requisição HTTP normal dura milissegundos e libera o servidor. Um WebSocket dura minutos ou horas: cada visitante ligado é uma conexão a ocupar memória e um «lugar» no processo. Dez visitantes são dez conexões abertas ao mesmo tempo, mesmo que ninguém escreva nada. Daí a diferença entre onde cabe e onde não cabe.

	No cPanel (compartilhada)	Num VPS
Conexões abertas	Ocupam recursos limitados da conta; um servidor de websockets residente choca com o limite de processos.	Só as limita a memória e o processador do VPS.
Quem garante o proxy	O Passenger. Não temos como prometer que as conexões longas passem de forma estável.	Você, com o nginx: tem de passar os cabeçalhos de *upgrade*.
Socket.IO	Pode iniciar por *polling* (pedidos repetidos) quando o WebSocket não passa. Funciona, mas é um pedido atrás do outro.	Usa o WebSocket de verdade.
Recomendação	Teste com poucos usuários; para tempo real de verdade, mude.	O lugar certo.

Num VPS com nginx, passo a passo

1. **Ponha a aplicação a ouvir só no próprio servidor** (`127.0.0.1`) e a rodar sempre, com o PM2 ou o systemd: **manter a aplicação a rodar com o PM2**.

2. **No bloco do nginx**, acrescente as linhas que deixam passar a mudança de HTTP para WebSocket:

```
location / {
    proxy_pass http://127.0.0.1:3000;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Host $host;
    proxy_read_timeout 3600s;
}
```

   O `proxy_read_timeout` importa: por padrão o nginx fecha uma conexão que fique 60 segundos sem dados, e os websockets inativos caem. O valor acima é um exemplo.

3. **No navegador, use** `wss://` (e não `ws://`) quando o site é HTTPS, no mesmo domínio. Um site em HTTPS que abra `ws://` é bloqueado pelo navegador.

4. **Recarregue o nginx** depois de validar a configuração: `sudo nginx -t` e `sudo systemctl reload nginx`.

5. **Confirme no navegador:** nas ferramentas de programador, aba «Rede», filtre por «WS». Uma conexão com o estado «101» é um WebSocket a funcionar.

Se o Socket.IO «funciona» mas é lento

, pode estar caindo para *polling* sem que ninguém repare: a aba «Rede» mostra pedidos repetidos em vez de uma conexão `101`. É sinal de que o WebSocket não está passando, quase sempre por falta das linhas de *upgrade* no proxy.

Várias instâncias da aplicação?

Com mais de um processo atrás do proxy, os clientes do Socket.IO precisam de ficar sempre no mesmo processo ou de um adaptador compartilhado: a documentação do projeto explica. Comece com um processo. Num VPS não gerenciado, a manutenção é sua: **até onde vai o nosso suporte**.

RECOMENDADO

Precisa de conexões em tempo real? Um VPS dá-lhe o controle do proxy e dos processos.

Ver os servidores VPS →

Onde ver na sua conta

Abra o **Uso de recursos** (Resource Usage) no cPanel: ele mostra os processos, a memória e a CPU da sua conta contra os limites, e marca quando um limite foi atingido. Quando o limite de processos é atingido, os novos pedidos são recusados e os visitantes podem ver o erro 508; quando a memória acaba, as coisas falham. Um servidor de websockets residente mantém um processo e sua memória ocupados enquanto houver alguém conectado, então chega rápido a esse teto. Para conexões em tempo real o caminho é um VPS, que dá acesso total por SSH e controle do nginx, ou um servidor Cloud. O VPS1 (1 vCPU, 2 GB de RAM, 40 GB NVMe) ou o Cloud1 (1 vCPU, 2 GB, 25 GB) é um bom ponto de partida, e você pode subir até o VPS6 ou o Cloud6 (32 vCPU, 64 GB) conforme as conexões crescem.

VEJA TAMBÉM

- **Nginx como proxy reverso à frente de um contêiner** →
- **Manter uma aplicação Node.js a rodar num VPS com o PM2** →
- **Portas e firewall num VPS: porque a aplicação não é alcançável** →

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-3402 · v2026.10.18 · Gerado a 14/11/2026 22:20