interweb

Generated on 15/11/2026 00:13

KNOWLEDGE BASE · PUBLIC DOCUMENT

# CORS error between your front end and your API: how to fix it

Node.js & JavaScript

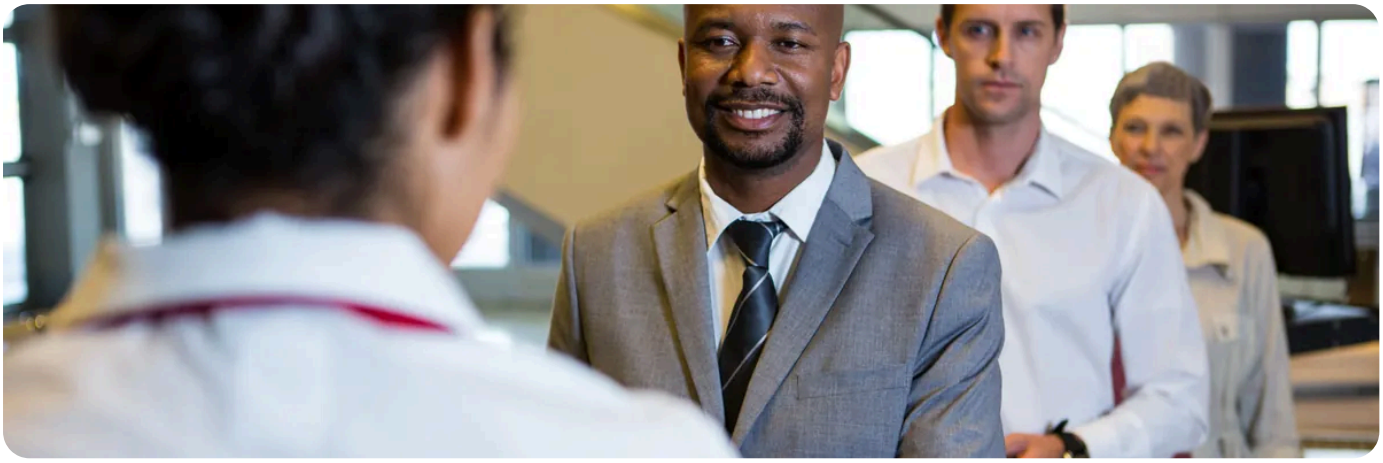
| CODE | VERSION | COUNTRY · LANGUAGE |
| --- | --- | --- |
| RLX-AO-KB-1082 | v2026.10.10 | Angola · English |

**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/1082

# CORS error between your front end and your API: how to fix it

Node.js & JavaScript  Created on **October 10, 2026**  2 min read



*"Access to fetch at … has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present"*. It is one of the most searched messages for anyone who splits the site (React, Vue) from the API (Node, Python). **The short answer:** the block is done by the **browser**, not the server, and it is lifted only if the **API** says, in response headers, that this site's origin may read the response. The request reached the API; it is the browser that will not let your JavaScript see the result.

## What an "origin" is

The origin is the set of **protocol + domain + port**. `https://asuaempresa.ao`, `https://www.asuaempresa.ao`, `http://asuaempresa.ao` and `https://api.asuaempresa.ao` are four different origins. If the site is served from one and the API from another, it is a cross-origin request, and the browser demands permission. For requests that change data or carry custom headers, it first sends a "preflight" request (`OPTIONS`); if the API does not answer that one properly, the real request never leaves.

## Making the API allow the right origin

| The API is in… | Library | Configuration (example) |
|---|---|---|
| **Express (Node.js)** | `cors` (`npm install cors`) | `app.use(cors({ origin: "https://asuaempresa.ao" }))` |
| **Flask** | `flask-cors` | `CORS(app, origins=["https://asuaempresa.ao"])` |
| **Django** | `django-cors-headers` | `CORS_ALLOWED_ORIGINS = ["https://asuaempresa.ao"]` and the *middleware* at the top of the list |
| **FastAPI** | Ships with `CORSMiddleware` | `app.add_middleware(CORSMiddleware, allow_origins=["https://asuaempresa.ao"])` |

Give the **exact** origin of your site, with the protocol and no trailing slash. Then restart the API: **Node applications**, and Python ones, keep the old code until they are restarted.

**The CORS error can be hiding another one.**

If the API returns a 500 or a 503, the error page carries no CORS headers, and the browser shows the CORS error instead of the real one. Before touching the configuration, open the API address directly (or use `curl -i`) and see whether it answers properly. If it does not, the problem is not CORS: **a 503 on a Node.js app**.

**Do not put `*` out of laziness.**

Allowing any origin is acceptable for a public read-only API. For an API with accounts, passwords or cookies it is not: it opens the door to any site. On top of that, the browser rejects `*` when the request carries cookies or credentials. List the origins you actually use.

**The solution that avoids the problem:**

Serve the site and the API from the *same origin*, for instance the API on an `/api` path of the same domain, behind a proxy. With no cross-origin requests there is no CORS. On a VPS it comes naturally: **nginx as a reverse proxy**.

NEED A HAND?

The API answers curl and the browser still blocks it? Tell us the site's domain, the API's, and what the browser console shows.

Open a support ticket →

SEE ALSO

- **Publishing a React or Vue site: the build folder, and the refresh that gives a 404** →
- **Running a Flask or FastAPI application** →
- **Nginx as a reverse proxy in front of a container** →

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<br>Belas, Luanda, Angola |
| CONTACT | info@interweb.ao · +244 942 011 191 · www.interweb.ao |
| OPENING HOURS | Monday to Friday - 08 AM to 04 PM<br>Saturday & Sunday - Closed |

RLX-AO-KB-1082 · v2026.10.10 · Generated on 15/11/2026 00:13

Online version of this document