# Copilot: CSRF Django cookie Error

**Prompt**

django restAPI server is created, I always get that error. Client app run in cli with axios. The client app does run in localhost:port. The error: Forbidden (CSRF cookie not set.)

Sounds like you're running into one of Django's most notorious headaches: CSRF protection woes when working with axios and a separate frontend. Let's untangle that.

**🚧 What's Happening**

Django's CSRF protection relies on setting a special cookie (csrftoken) and expecting that token to be sent back in the request headers. But if your frontend is running separately via CLI (say React or Vue on localhost:3000), Django might not send or accept those cookies unless you’ve configured things just right.

**🛠 Fixes and Setup Checklist**

Here’s a breakdown to help you get past the Forbidden (CSRF cookie not set.) error:

**✅ Backend (Django)**

Make sure you have **CSRF and CORS configured properly** in settings.py:

# settings.py

INSTALLED\_APPS = [

...

'corsheaders',

...

]

MIDDLEWARE = [

"corsheaders.middleware.CorsMiddleware", # Must be before CommonMiddleware

"django.middleware.common.CommonMiddleware",

"django.middleware.csrf.CsrfViewMiddleware",

...

]

CORS\_ALLOWED\_ORIGINS = [

"http://localhost:3000", # Or whatever your frontend port is

]

CORS\_ALLOW\_CREDENTIALS = True

CSRF\_TRUSTED\_ORIGINS = [

"http://localhost:3000"

]

CSRF\_COOKIE\_SAMESITE = 'None'

CSRF\_COOKIE\_SECURE = True # Only if using HTTPS

**✅ Frontend (axios setup)**

// axios config

axios.defaults.withCredentials = true;

axios.defaults.xsrfCookieName = "csrftoken";

axios.defaults.xsrfHeaderName = "X-CSRFToken";

Make sure you're doing a **GET** request first so the browser receives the CSRF cookie, then follow up with your POST.

You can also manually extract the token from cookies and inject it:

function getCookie(name) {

let cookieValue = null;

if (document.cookie && document.cookie !== "") {

const cookies = document.cookie.split(";");

for (let cookie of cookies) {

cookie = cookie.trim();

if (cookie.startsWith(name + "=")) {

cookieValue = decodeURIComponent(cookie.slice(name.length + 1));

break;

}

}

}

return cookieValue;

}

const csrfToken = getCookie("csrftoken");

axios.post("http://localhost:8000/api/", data, {

headers: {

"X-CSRFToken": csrfToken

},

withCredentials: true

});

**❓ Still Stuck?**

* Is the CSRF token actually being sent in the header? Use browser dev tools to check.
* Are cookies being blocked by CORS settings?
* Are you accessing the site over HTTPS with CSRF\_COOKIE\_SECURE = True? This won’t work over plain HTTP.

If you want to walk through your exact setup together, just share a snippet of your server and client config and I’ll help patch the gaps. Let’s squash this bug. 🐛