

# API Debugging Guide: Next.js + Django + Docker

## Basic Debugging Flow

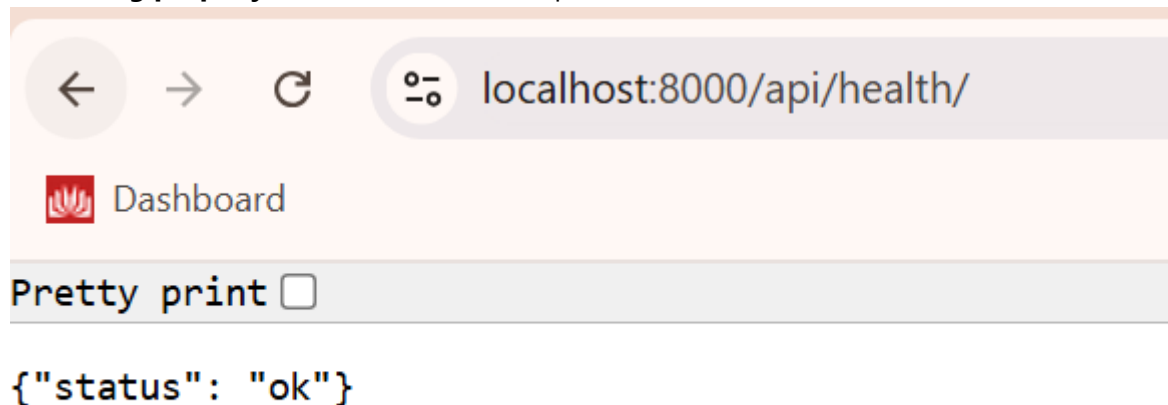
### Step 1: Check Backend Directly

Open your browser and navigate directly to your API endpoint:

```
https://localhost:8000/api/health
https://localhost:8000/api/users/login/
...
```

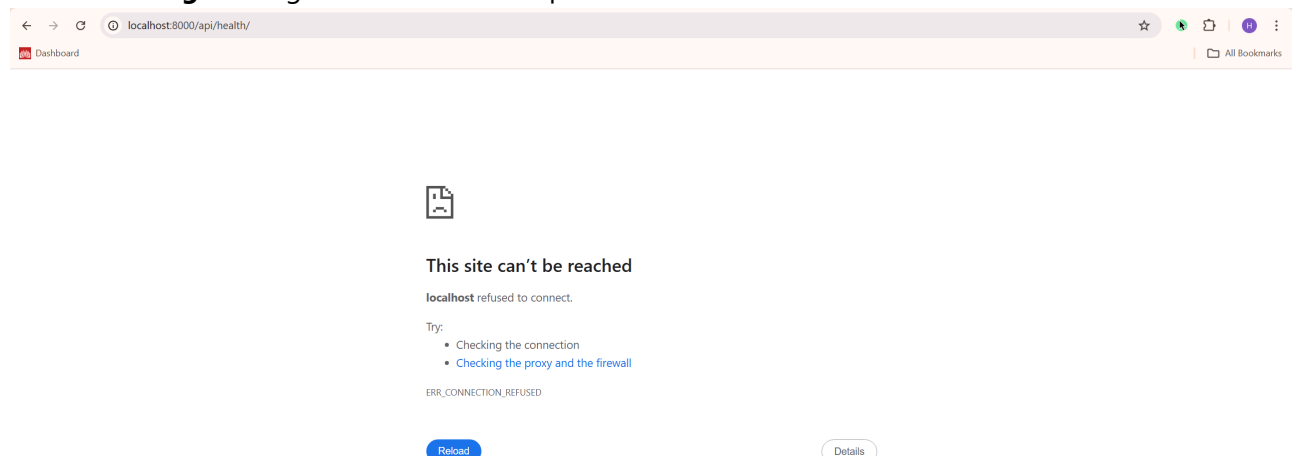
### Step 2: Evaluate the Backend Response

- **If working properly:** You'll see a normal response



Proceed to step 3b check frontend

- **If not working:** You'll get an error or no response

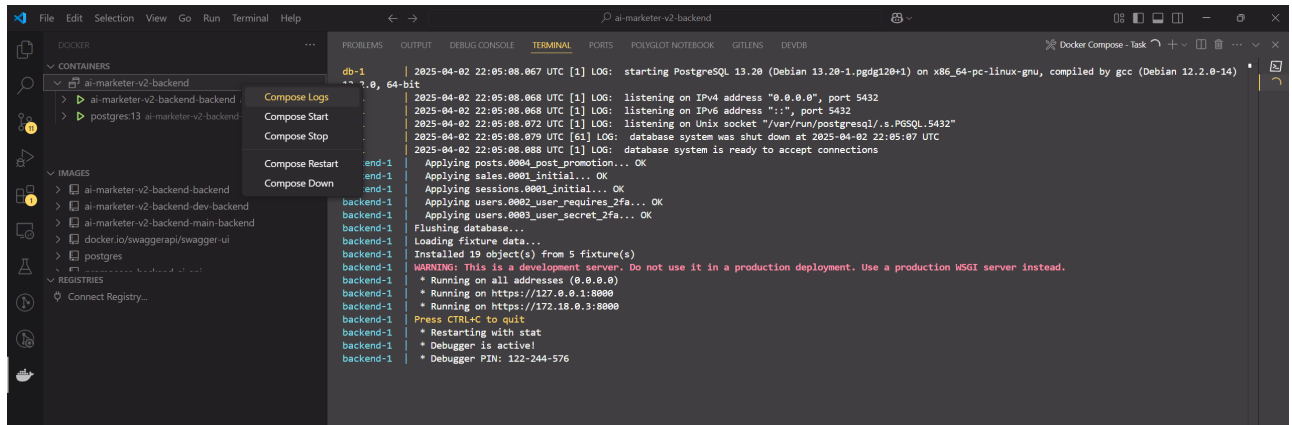


-> Proceed to step 3a Check Docker logs

### Step 3a: Check Docker Logs (if backend doesn't work)

In VSCode:

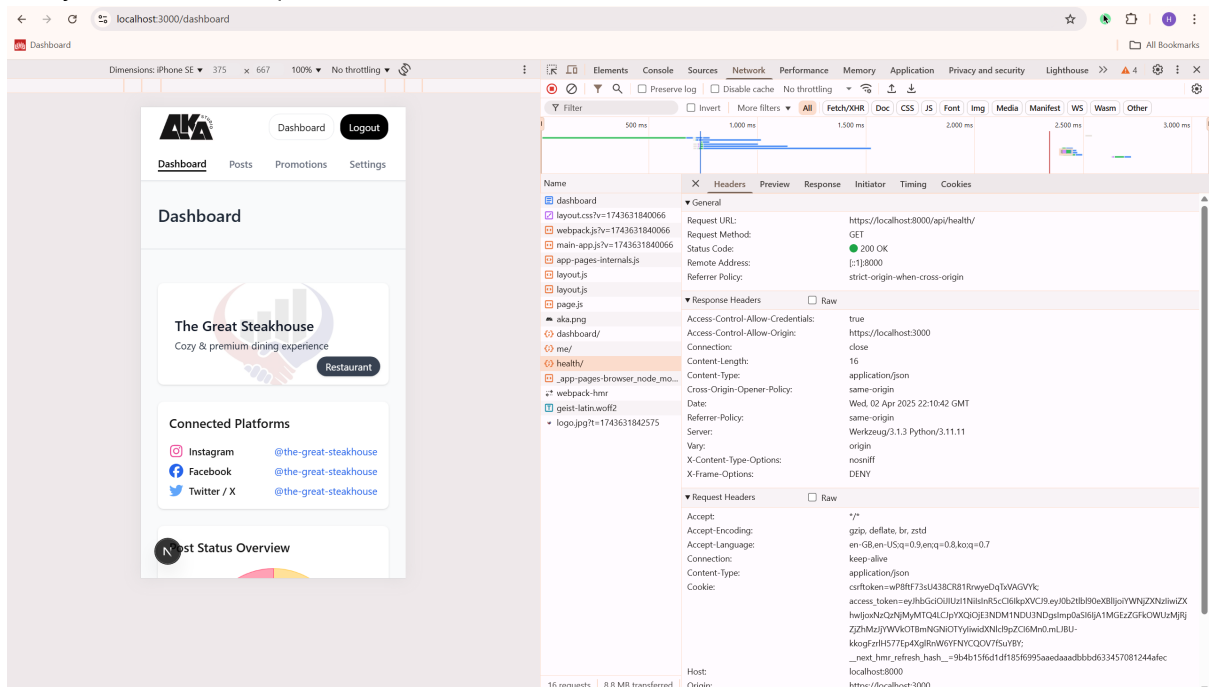
1. Select Docker extension tab
2. Find your container
3. Right-click → Compose Logs



Look for error messages that explain why the backend isn't responding. If there are too many logs to search through, try accessing the backend directly in your browser while monitoring the logs simultaneously.

### Step 3b: Check Frontend Request (if backend works)

1. Open your Next.js app in browser
2. Open DevTools (F12)
3. Go to Network tab
4. Look at the API requests:
  - Check if the URL in the request matches the one you accessed directly
  - Verify there are no request errors



- For example, if you're seeing an error with "health/" endpoint, find and click on that request in the Network tab. This will open a details panel showing General, Response Headers, and Request Headers sections. In the General section, verify that the Request URL matches exactly what you've tried to access directly in the browser.

- For authenticated pages (like login, dashboard, etc.), if you see a 401 Unauthorized error, check the Request Headers section and look for the "Cookie" field to verify if authentication cookies are properly included in the request. Missing or invalid cookies often cause authentication failures.

## Step 4: Debug Frontend Code (if needed)

If the frontend request doesn't match what's expected:

- Add console.log statements starting from the component making the API call
- Work your way up through parent components to find where the issue originates
- Check environment variables and API URL construction

## Common Issues

### 1. Docker container not running properly

- Check container status with `docker ps`
- Restart with `docker compose up -d --build` after `docker compose down -v`

### 2. Wrong API URL in Next.js

- Environment variables misconfigured
- Using localhost instead of service name in Docker
- Path or endpoint typos

### 3. CORS errors

- Missing CORS configuration in Django
- Frontend origin not allowed in backend

### 4. Authentication issues

- Missing or expired tokens
- Incorrect headers