Webhook Troubleshooting Guide

Simple Explanation: What Went Wrong & How to Avoid It ■

Imagine you're trying to send a letter to your friend's house, but you wrote the wrong house number on the envelope. The mail carrier goes to the wrong house, nobody's home there, and your letter gets returned. That's exactly what happened here!

What Was the Error? ■

The Problem:

- Your **backend server** (FastAPI) was running on **port 8000** (like house #8000)
- But **ngrok** was trying to forward traffic to **port 80** (like house #80)
- When Clerk sent webhooks, they reached ngrok fine, but ngrok knocked on the wrong door (port 80) where nobody was home!
- Result: **502 Bad Gateway** = "I found the street, but nobody's at this house number"

Think of it like this:

- **Backend Server** = Your house (where you actually live)
- **Port Number** = Your house number
- **ngrok** = A mail forwarding service
- **Clerk Webhooks** = Letters being sent to you

If the forwarding service has the wrong house number, the letters never reach you!

What Were the TWO Issues?

Issue #1: CORS Blocking (403 Forbidden) ■

Problem: Your backend only accepted requests from localhost:5173 (your frontend)

Why it failed: Clerk webhooks come from Clerk's servers, not your frontend!

The Fix:

```
# Before (too strict):
allow_origins=["http://localhost:5173"]  # Only frontend allowed

# After (open to webhooks):
allow_origins=["*"]  # All origins allowed (safe because Svix verifies signatures)
```

Real-world analogy: Your security guard (CORS) was told "only let in people from the Smith family." But the delivery driver (Clerk webhook) isn't a Smith, so they got turned away!

Issue #2: Wrong Port (502 Bad Gateway) ■

Problem: ngrok forwarding to port 80, but backend running on port 8000

The Fix:

```
# Wrong:
ngrok http 80  # Nobody's home at port 80!

# Correct:
ngrok http 8000  # Backend is actually here
```

How to Avoid This in the Future ■■

■ Checklist Before Starting ngrok:

1. Know Your Port Numbers:

```
# Backend: Always port 8000
python -m uvicorn app.main:app --reload --port 8000

# Frontend: Always port 5173
npm run dev
```

2. Start ngrok with the RIGHT Port:

```
# For backend webhooks:
ngrok http 8000  ■

# Common mistake:
ngrok http 80   ■ (wrong!)
ngrok http 5173 ■ (that's for frontend, not backend!)
```

3. Remember the Order:

- Step 1: Start backend (port 8000)
- Step 2: Start ngrok (pointing to 8000)
- Step 3: Copy ngrok URL
- Step 4: Update Clerk webhook URL
- Step 5: Test!

4. Quick Test Command:

```
# After starting ngrok, test it works:
curl https://your-ngrok-url.ngrok-free.dev/health

# Should return: {"status":"healthy"}
# If it fails, your port is wrong!
```

■ Port Number Quick Reference

Service	Port	Command
Backend (FastAPI)	8000	uvicorn app.main:app --port 8000
Frontend (Vite)	5173	npm run dev
ngrok (for backend)	8000	ngrok http 8000

ngrok (for frontend)	5173	ngrok http 5173

■ Mental Model

Think of it like a **phone system**:

```
Internet (Clerk)
    ↓
ngrok (receptionist) - "Let me forward your call..."
    ↓
Port 8000 (your desk) - "Hello, I'm here!"
    ↓
Backend (you) - "Got the message!"
```

If ngrok dials **extension 80** instead of **extension 8000**, the call never reaches you!

Common Mistakes & How to Spot Them ■

Mistake 1: Wrong Port

Symptom: 502 Bad Gateway or ERR_NGROK_8012

Fix: Check ngrok command - should be 'ngrok http 8000'

Mistake 2: Backend Not Running

Symptom: Same as above!

Fix: Make sure uvicorn is running in another terminal

Mistake 3: CORS Blocking

Symptom: 403 Forbidden in ngrok

Fix: Use allow_origins=["*"] for webhooks (already fixed!)

Mistake 4: Forgot to Update Clerk URL

Symptom: Webhooks never arrive

Fix: Go to Clerk Dashboard → Webhooks → Update URL with new ngrok URL

Pro Tips ■

1. Keep a notes file with your commands:

```
# my-commands.txt
cd backend && python -m uvicorn app.main:app --reload --port 8000
ngrok http 8000
```

2. Check ngrok dashboard at http://127.0.0.1:4040

- Shows all requests in real-time
- If you see 502, port is wrong
- If you see 200, it's working!

3. One ngrok per terminal:

- Terminal 1: Backend server (port 8000)
- Terminal 2: ngrok (forwarding to 8000)
- Terminal 3: Frontend server (port 5173) - optional

4. Test before using:

```
# Start backend
curl http://localhost:8000/health  # Should work

# Start ngrok
curl https://your-ngrok-url/health  # Should also work

# Now update Clerk!
```

Complete Webhook Setup Guide ■

Step-by-Step Process

1. Start Your Backend Server

```
cd backend
python -m uvicorn app.main:app --reload --port 8000
```

Expected output:

```
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process
```

2. Start ngrok (New Terminal)

```
ngrok http 8000
```

Expected output:

```
Session Status                online
Forwarding                    https://your-unique-url.ngrok-free.dev -> http://localhost:8000
```

■■ Important: Copy the HTTPS forwarding URL!

3. Test ngrok Connection

```
curl https://your-unique-url.ngrok-free.dev/health
```

Expected response:

```
{"status":"healthy"}
```

■ If this fails, your port number is wrong!

4. Update Clerk Webhook URL

1. Go to Clerk Dashboard (https://dashboard.clerk.com)
2. Navigate to **Webhooks** in the sidebar
3. Click on your webhook endpoint
4. Update **Endpoint URL** to: https://your-unique-url.ngrok-free.dev/api/webhooks/clerk
5. Make sure these events are enabled: user.created, user.updated, user.deleted
6. Save changes

5. Test the Webhook

1. Go to your Clerk Dashboard → **Users**
2. Delete any test users (if they exist)
3. Sign up with a new test account in your app
4. Check ngrok dashboard at http://127.0.0.1:4040
5. Should see: POST /api/webhooks/clerk with **200 OK**

6. Verify in Supabase

1. Go to Supabase Dashboard (https://supabase.com/dashboard)
2. Navigate to **Table Editor** → users table
3. Your new user should appear with: clerk_user_id, email, full_name, created_at timestamp

Error Messages Decoded ■

Error	What It Means	Solution
502 Bad Gateway	ngrok can't reach your backend	Check port number (ngrok http 8000)
ERR_NGROK_8012	Connection refused at localhost	Backend not running OR wrong port
403 Forbidden	CORS blocking the request	Change allow_origins to ["*"]
401 Unauthorized	Webhook signature invalid	Check CLERK_WEBHOOK_SECRET in .env
404 Not Found	Wrong endpoint URL	Should be /api/webhooks/clerk

Quick Debugging Checklist ■

When webhooks fail, check these in order:

■ Backend is running on port 8000

■ ngrok is forwarding to port 8000 (not 80!)

■ ngrok URL has been copied correctly

■ Clerk webhook URL has been updated

■ Clerk webhook secret matches .env file

■ CORS allows all origins (allow_origins=["*"])

■ Can access /health endpoint via ngrok URL

Summary in 3 Sentences ■

1. **Always match ngrok's port to where your backend is actually running** (port 8000, not 80!)

2. **CORS must allow webhook sources** (Clerk servers), not just your frontend

3. **Test the connection** with curl before updating Clerk webhook URL

> **■ Remember: Port numbers are like house addresses - if you tell someone the wrong address, they can't find you! Always double-check your port numbers before starting ngrok. ■■**

Additional Resources ■

- **ngrok Documentation**: *https://ngrok.com/docs*
- **Clerk Webhooks Guide**: *https://clerk.com/docs/integrations/webhooks/overview*
- **FastAPI CORS Documentation**: *https://fastapi.tiangolo.com/tutorial/cors/*
- **Supabase RLS Guide**: *https://supabase.com/docs/guides/auth/row-level-security*

Project: TEAMFIT MVP - Team Building Platform
Last Updated: January 2025