

NGINX Setup Guide for QuickStartCommerce

This document explains how to configure NGINX as a reverse proxy to serve your frontend, backend APIs, and admin panel using a clean domain and subpaths like `/api` and `/admin`.

Table of Contents

1. Introduction
 2. Prerequisites
 3. Ubuntu/Debian Setup
 4. macOS Setup
 5. Windows Setup (with WSL2)
 6. Environment Configuration
 7. Troubleshooting
-

1. Introduction

Your QuickStartCommerce project includes:

- Frontend (Next.js - runs on port `3000`)
- Backend APIs & Admin Panel (Express.js - runs on port `5000`)

We'll use NGINX to map:

- `yourdomain.com` → Frontend
- `yourdomain.com/api` → Backend APIs
- `yourdomain.com/admin` → Admin Panel

2. Prerequisites

- Your server is running Ubuntu/Debian/macOS/Windows with WSL2
 - Node.js and PM2 installed
 - Your domain is already pointed to your server's IP (via DNS)
 - Backend running on port 5000, frontend on 3000
 - MongoDB URI and Stripe keys set up (see separate docs)
-

3. NGINX Setup on Ubuntu/Debian

Step 1: Install NGINX

```
sudo apt update
sudo apt install nginx -y
```

Step 2: Edit NGINX Config

```
sudo nano /etc/nginx/sites-available/yourdomain.com
```

Replace the contents with:

```
server {
    listen 80;
    server_name yourdomain.com;

    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

```
location /api {
    proxy_pass http://localhost:5000/api;
    proxy_http_version 1.1;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
}

location /admin {
    proxy_pass http://localhost:5000/admin;
    proxy_http_version 1.1;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
}
}
```

Step 3: Restart NGINX

```
sudo ln -s /etc/nginx/sites-available/yourdomain.com /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl restart nginx
```

4. NGINX Setup on macOS (Using Homebrew)

Step 1: Install NGINX

```
brew install nginx
```

Step 2: Edit NGINX Config

```
nano /opt/homebrew/etc/nginx/nginx.conf
```

Step 3: Add Server Block (similar to Ubuntu section)

Step 4: Restart NGINX

```
brew services restart nginx
```

5. NGINX Setup on Windows (WSL2 + Ubuntu)

- Open Ubuntu (WSL2) and follow the **Ubuntu steps above**
 - WSL2 will bind to localhost, and you can access it via browser on Windows
 - Make sure ports 80, 3000, and 5000 are allowed in Windows Firewall
-

6. Environment Configuration

Update your `.env` files with production-ready values:

```
# .env
MONGO_URI=your_mongodb_uri
JWT_SECRET=your_jwt
STRIPE_SECRET=your_key
EMAIL_USER=your_gmail@gmail.com
EMAIL_PASS=your_app_password
CLIENT_URL=https://yourdomain.com
```

For frontend:

```
NEXT_PUBLIC_ADMIN_BASE_URL=https://yourdomain.com
```

7. Troubleshooting

- **Port already in use:** Kill existing processes or change port
- **NGINX not forwarding:** Check if backend is accessible via `localhost:5000`
- **CORS issues:** Ensure backend includes proper CORS headers
- **WSL DNS errors:** Restart LxssManager in Windows Services