# Shopify Crypto Wallet Integration - Complete Setup Guide

## Project Overview

This Shopify app adds crypto wallet address collection during checkout and shows celebration modals after order completion. Based on the provided code, here's how to set it up with your own credentials.

## Prerequisites

- Node.js (v14+)
- npm or yarn
- Shopify Partner account
- ngrok or similar tunneling service (for webhooks)

## Step 1: Shopify Partner Setup

### 1.1 Create Shopify Partner Account

1. Go to https://partners.shopify.com
2. Create a free Partner account
3. Verify your account

### 1.2 Create Development Store

1. In Partner Dashboard → "Stores" → "Add store"
2. Choose "Development store"
3. Set up your test store (remember the `.myshopify.com` URL)

### 1.3 Create Shopify App

1. Partner Dashboard → "Apps" → "Create app"
2. Choose "Public app"
3. Fill in app details:
   - **App name**: "Crypto Wallet Integration"
   - **App URL**: `https://your-ngrok-url.ngrok.io` (we'll get this in Step 2)
   - **Allowed redirection URLs**: `https://your-ngrok-url.ngrok.io/auth/callback`

### 1.4 Get App Credentials

After creating the app, copy:

- **API Key** (Client ID)
- **API Secret Key** (Client Secret)

## Step 2: Setup ngrok (Required for Webhooks)

### 2.1 Install ngrok

```bash
# Download from https://ngrok.com/download
# Or install via npm
npm install -g ngrok

# Or via homebrew (Mac)
brew install ngrok
```

### 2.2 Start ngrok

```bash
# Start ngrok for port 3001 (backend port)
ngrok http 3001

# Copy the https URL (e.g., https://abc123.ngrok.io)
```

### 2.3 Update Shopify App URLs

Go back to Partner Dashboard → Your App → "App setup":

- **App URL**: `https://your-ngrok-url.ngrok.io` (your ngrok HTTPS URL)
- **Allowed redirection URLs**: `https://your-ngrok-url.ngrok.io/auth/callback`

## Step 3: Environment Configuration

### 3.1 Backend .env Setup

Create `.env` file in your **backend** directory:

env

```
# Shopify App Credentials (from Step 1.4)
SHOPIFY_API_KEY=your_api_key_from_partner_dashboard
SHOPIFY_API_SECRET=your_api_secret_from_partner_dashboard

# URLs (use your ngrok HTTPS URL)
HOST=https://your-ngrok-url.ngrok.io
FRONTEND_URL=http://localhost:3000

# App Permissions
SCOPES=read_orders,write_orders,read_products,write_products,read_script_tags,write_script_tags

# JWT Secret (generate random 32+ character string)
JWT_SECRET=your_super_secure_random_jwt_secret_min_32_characters

# Server Port
PORT=3001
```

**Generate JWT Secret:**

bash

```
# Use this command to generate secure JWT secret
node -e "console.log(require('crypto').randomBytes(32).toString('hex'))"
```

### 3.2 Frontend Environment (if needed)

Create `.env` in **frontend** directory:

env

```
REACT_APP_API_URL=http://localhost:3001
```

## Step 4: Webhook Configuration

### 4.1 Configure Webhooks in Shopify Partner Dashboard

1. Go to Partner Dashboard → Your App → "App setup"
2. Scroll to "Webhooks" section
3. Add these webhooks with your ngrok HTTPS URL:

**Order Creation Webhook:**

- **Event**: `orders/create`
- **URL**: `https://your-ngrok-url.ngrok.io/webhooks/orders/create`
- **Format**: JSON

**App Uninstalled Webhook:**

- **Event**: `app/uninstalled`
- **URL**: `https://your-ngrok-url.ngrok.io/webhooks/app/uninstalled`
- **Format**: JSON

### 4.2 Required App Permissions

In "App setup" → "App permissions", ensure these are enabled:

- ✅ **Orders**: Read and write
- ✅ **Products**: Read and write
- ✅ **Script tags**: Read and write
- ✅ **Checkout**: Read and write (if available)

## Step 5: Install Dependencies

### 5.1 Backend Dependencies

```bash
# Navigate to backend directory
cd backend
npm install

# Key packages should include:
# - express
# - cors
# - helmet
# - body-parser
# - cookie-parser
# - crypto
# - jsonwebtoken
# - @shopify/shopify-api
# - dotenv
# - node-fetch
```

### 5.2 Frontend Dependencies

bash

```
# Navigate to frontend directory
cd frontend
npm install

# Key packages should include:
# - react
# - react-router-dom
# - axios
# - react-hot-toast
# - lucide-react
```

## Step 6: Run the Application

### 6.1 Start Backend Server

bash

```
# In backend directory
npm start
# or
node server.js

# Should see:
# 🚀 Server running on port 3001
# 📱 Frontend URL: http://localhost:3000
# 🌐 Host URL: https://your-ngrok-url.ngrok.io
# 🔑 Shopify API Key: your_api_key
```

### 6.2 Start Frontend Application

bash

```
# In frontend directory
npm start

# React app starts on http://localhost:3000
```

### 6.3 Keep ngrok Running

Make sure ngrok is still running on port 3001 in a separate terminal.

## Step 7: Test Installation

### 7.1 Install App on Development Store

1. Open browser → `http://localhost:3000`
2. Enter your development store name (without .myshopify.com)
3. Click "Install App"
4. Authorize permissions on Shopify
5. Should redirect back to dashboard

### 7.2 Test Checkout Integration

1. Go to your development store frontend
2. Add product to cart → checkout
3. Look for "Crypto Wallet Address" field
4. Complete test order
5. Should see celebration modal

## Step 8: Code Structure Overview

Based on your provided code:

```
project/
├── backend/
│   ├── server.js          # Main Express server
│   ├── package.json
│   └── .env              # Your environment variables
├── frontend/
│   ├── src/
│   │   ├── App.js        # Main React app with routing
│   │   └── components/
│   │       ├── Dashboard.js     # Main dashboard
│   │       ├── Installation.js  # OAuth installation flow
│   │       ├── Settings.js      # App settings
│   │       ├── Orders.js        # Orders with wallet data
│   │       ├── Header.js        # App header
│   │       └── CelebrationModal.js # Success modal
│   └── package.json
```

## Step 9: Key Features Implemented

Based on your code, the app includes:

✅ **OAuth Installation Flow** - Automatic Shopify app installation
✅ **Checkout Script Injection** - Adds wallet field to checkout
✅ **Order Tracking** - Tracks orders with wallet addresses
✅ **Celebration Modal** - Shows success modal after orders
✅ **Dashboard** - Stats and recent wallet orders
✅ **Settings Panel** - Configure app behavior
✅ **Webhook Handling** - Processes order creation and app uninstall

## Step 10: Troubleshooting

### Common Issues:

#### "Shop parameter is required"

- Ensure store name entered correctly (no .myshopify.com)

#### "Authentication failed"

- Check API Key/Secret in .env
- Verify callback URL matches ngrok URL exactly

#### Webhook not receiving data

- Ensure ngrok is running and HTTPS URL is used
- Check webhook URLs in Partner Dashboard
- Verify HMAC verification in code

#### Wallet field not showing in checkout

- Check browser console for errors
- Verify script tag installation
- Test in incognito mode

#### Frontend can't connect to backend

- Ensure both servers are running
- Check CORS configuration
- Verify API URLs

### Debug Tips:

1. Check browser Network tab for failed requests
2. Check server console logs for errors
3. Use Shopify Partner Dashboard → Apps → Your App → "Debug" for app logs
4. Test webhooks using tools like Postman

## Step 11: Going Live

### For Production Deployment:

1. Deploy backend to cloud service (Heroku, Railway, etc.)
2. Deploy frontend to hosting service (Vercel, Netlify, etc.)
3. Update .env with production URLs
4. Update Shopify App URLs to production URLs
5. Update webhook URLs to production backend
6. Test thoroughly before submitting to App Store

## Security Notes

- Never commit .env files to git
- Use HTTPS for all production URLs
- Validate all webhook requests
- Implement proper error handling
- Use secure JWT secrets

## Support Resources

- Shopify App Development Docs
- Shopify API Reference
- ngrok Documentation

---

**Important**: Keep ngrok running during development as webhooks need the HTTPS tunnel to work properly. The app uses session storage in memory - consider using a database for production.