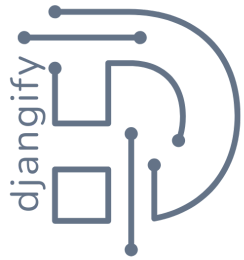

How to Add Stripe to Your Self-Hosted eCommerce Builder

This guide explains how to connect Stripe to your **self-hosted eCommerce Builder** so you can accept payments for **digital downloads**.

In the self-hosted version:

- All products and prices are created inside the eCommerce Builder
- Stripe is used **only** to take payments and confirm them securely
- No pricing is managed inside Stripe
- Stripe configuration is done through the ShopSettings Dashboard and/or your .env file
- Email configuration is done through the ShopSettings Dashboard and/or your .env file

---

What You Need Before You Start

You must have:

- A Stripe account
- Your eCommerce Builder already running
- HTTPS enabled on your domain
- Access to your server and .env file
- Admin access to /admin

How Payments Work

Understanding this helps avoid misconfiguration:

1. You create products and prices **inside the eCommerce Builder**
2. At checkout, the system calculates the total internally
3. A Stripe **PaymentIntent** is created dynamically
4. Stripe collects card details securely
5. Stripe confirms the payment via a webhook
6. The eCommerce Builder:
    - Creates and stores the order
    - Unlocks the downloads
    - Sends the order confirmation email

Using Stripe in Test Mode (Strongly Recommended)

Before accepting real payments, you should test your store using **Stripe Test Mode**. Test Mode lets you simulate payments without charging real cards or money.

What Test Mode Is Used For

Stripe Test Mode allows you to verify that:

- Checkout loads correctly
- Payments complete successfully
- Orders are created in the admin
- Downloads unlock correctly
- Webhooks are firing and being processed

No real charges are made.

Step 1: Switch Stripe to Test Mode

1. Log in to Stripe: https://dashboard.stripe.com
2. Go to the left hand menu with your business name (drop down) and select **Switch To Sandbox** (top-right of the dashboard)
3. You will now see **test data only**

Step 2: Get Your Test API Keys

While still in Test Mode:

1. Go to **Developers → API keys**
2. Copy:
    - **Publishable key** (`pk_test_...`)
    - **Secret key** (`sk_test_...`)

These keys are **only for testing**.

Step 3: Create a Test Webhook Endpoint

Test mode requires its **own webhook**.

1. While in **Test Mode,** go to **Developers** *(bottom left-hand side of your screen)* → **Webhooks**
2. Click **Add destination**
3. Select events:
    - `payment_intent.succeeded`
    - `payment_intent.payment_failed`
4. Enter: `https://yourdomain.com/shop/webhook`
5. Create the endpoint
6. Copy the **Signing secret** (`whsec_...`)

Step 4: Update `.env` With Test Keys

Replace your Stripe values with **test keys**:

```
STRIPE_PUBLIC_KEY=pk_test_...
STRIPE_SECRET_KEY=sk_test_...
STRIPE_WEBHOOK_SECRET=whsec_...
```

Save the file.

Step 5: Restart the Application

After changing `.env`, restart eBuilder:

```
sudo systemctl restart ebuilder
```

Step 6: Add your email configuration info

There is a separate page on ***how to set up email confirmation for your self-hosted eCommerce store here.***

Step 7: Run Test Payments

Use Stripe's official test card:

```
Card number: 4242 4242 4242 4242
Expiry date: Any future date
CVC: Any 3 digits
Postcode: Any value
```

Test the full flow:

- Add product to cart
- Complete checkout
- Confirm:
    - Payment appears in Stripe (Test mode)
    - Order appears in Admin
    - Download access works

Step 8: Switch to Live Mode

Once testing is complete:

1. Switch Stripe back to **Live mode by clicking exit sandbox**
2. Replace `.env` values with **live keys**
3. Create a **live webhook endpoint** (same url you used in test mode)
4. Restart the application again

Your store is now ready for real customers.

Step 1: Get Your LIVE Stripe API Keys

1. Log in to Stripe
   https://dashboard.stripe.com
2. Make sure you are in **Live mode**
3. Go to **Developers → API keys**
4. Copy the following:
    - **Publishable key** (`pk_live_...`)
    - **Secret key** (`sk_live_...`)

Keep these safe. Do not share them.

Step 2: Create the Stripe Webhook (Required)

The webhook allows Stripe to notify your site when a payment succeeds or fails.

1. In Stripe, go to **Developers** → **Webhooks**
2. Click **Add endpoint**
3. Select **only** these events:
    - payment_intent.succeeded
    - payment_intent.payment_failed
4. Enter your webhook URL: https://yourdomain.com/shop/webhook
5. Create the endpoint
6. Copy the **Signing Secret** (whsec_...)

This secret is mandatory. You add this to your .env file and/or the dashboard area.

Step 3: Configure Your .env File

The self-hosted version **must** be configured via environment variables.

Your project includes a .env.example.
Copy it and create your real .env file.

Required Stripe Variables

```
STRIPE_PUBLIC_KEY=pk_live_...
STRIPE_SECRET_KEY=sk_live_...
STRIPE_WEBHOOK_SECRET=whsec_...
```

That is the complete Stripe configuration for the self-hosted shop.

No additional Stripe settings are required.

Step 4: Restart the Application

After saving your .env file, restart the application so the keys are loaded.

If using systemd / Gunicorn, restart the relevant service. E.g.

sudo systemctl restart ebuilder

Step 5: Verify Stripe Is Working

1. Log in to /admin
2. Create a test product with a small price
3. Visit your store and complete a real checkout
4. Confirm:
    - Payment appears in Stripe
    - Order appears in Admin → Orders
    - Download links are available
    - Confirmation email is sent

If all four work, Stripe is correctly configured. You can issue a refund if you want to test the process (optional)

Admin Stripe Fields (Self-Hosted)

Stripe fields may appear in **Admin → Dashboard Settings**.

For self-hosted installations:

- .env is the authoritative source
- Admin values are secondary and optional
- .env always takes precedence

This ensures predictable deployments and safe updates.

Common Problems and Fixes

Payments fail immediately

- Check STRIPE_SECRET_KEY
- Confirm you are in Live mode

Orders not created

- Webhook is missing or incorrect
- Check the webhook URL and signing secret

Nothing happens after payment

- Application was not restarted after .env change

Self-Hosted Summary

- One-off payments only
- Prices managed in the eCommerce Builder
- Stripe used strictly for payment processing
- No customer credit card or address details are collected
- .env configuration is required
- Webhook is mandatory

Stripe Pre-Launch Checklist

Use this checklist **before opening your store to customers**.

If every item is confirmed, Stripe payments and downloads should work correctly.

A. Stripe Account & Mode

- Stripe account is active and verified
- You have completed **Test Mode setup first**
- You have deliberately switched to **Live Mode** for launch
- You understand Test and Live mode are completely separate

B. Stripe API Keys

Confirm you are using **keys from the same mode** (this matters). **Never mix them.**

Test Mode (before launch)

- Publishable key starts with `pk_test_`
- Secret key starts with `sk_test_`

Live Mode (for launch)

- Publishable key starts with `pk_live_`
- Secret key starts with `sk_live_`

C. Webhook Endpoint (Critical)

Your shop webhook **must** be:

/shop/webhook/

Test Mode

- Webhook created while Stripe is in **Test Mode**
- Endpoint URL is correct and complete

  `https://YOURDOMAIN/shop/webhook/`
- Events selected:
  - `payment_intent.succeeded`
  - `payment_intent.payment_failed`
- Webhook signing secret copied (`whsec_...`)
- Test webhook secret saved in the store

Live Mode

- Webhook created again in **Live Mode**
- Endpoint URL is identical

  `https://YOURDOMAIN/shop/webhook/`
- Events selected:
  - `payment_intent.succeeded`
  - `payment_intent.payment_failed`
- Live webhook signing secret copied
- Live webhook secret saved in the store

Test and Live webhooks are separate. Both must exist.

D. Store Configuration

Self-Hosted

- Stripe keys are set in `.env`
- `.env` changes have been applied correctly:
    - Restart your application using your system service.
- Application restarted after **every** Stripe change

Managed Hosting

- Stripe keys saved in **Pages → Dashboard Settings**
- Keys were saved in the correct fields
- Page refreshed after saving

E. Frontend Checkout Verification

- Checkout page loads without errors
- Stripe card form appears correctly
- No console errors related to Stripe
- Payment button works and is clickable

F. End-to-End Payment Test

Test Mode (required)

- Test card `4242 4242 4242 4242` used
- Payment succeeds in Stripe Test Mode
- Order is created in admin
- Download access unlocks
- Stripe → Webhooks shows **Delivered (200)**

Live Mode (final confirmation)

- Small real payment completed
- Payment appears in Stripe Live Mode
- Order created in admin
- Downloads unlock correctly
- Webhook delivery successful

G. Final Sanity Checks

- HTTPS works on your domain
- Domain resolves publicly (no local-only URLs)
- Server firewall allows outbound HTTPS
- No maintenance mode enabled
- Emails (order confirmation) are sending

If **all boxes are ticked**, your Stripe setup is ready.

If the Webhook Does Not Fire (Troubleshooting)

If Stripe shows a successful payment but **no order is created**, the issue is almost always the webhook. Use this diagnostic path **in order**.

1. Check Stripe → Webhooks → Event Log

In Stripe Dashboard:

- Open the webhook endpoint
- Click the most recent event
- Look for:
    - **Status:** Delivered
    - **Response code:** 200

If you see:

- **No attempts** → Stripe never reached your site
- **4xx / 5xx** → your site rejected the request

2. Confirm the Webhook URL Exactly

It must be **exactly**:

https://YOURDOMAIN/shop/webhook/

Common mistakes:

- Missing trailing slash
- `/api/` instead of `/shop/`
- Typo in domain
- Using `http://` instead of `https://`

Stripe will not retry indefinitely if the URL is wrong.

---

3. Confirm Mode Alignment

This catches more errors than anything else.

- Test payment → Test webhook → Test secret
- Live payment → Live webhook → Live secret

If any one of these is mismatched, signature verification will fail silently.

---

4. Confirm Webhook Signing Secret

- Make sure the `whsec_...` value:
    - Matches the correct mode
    - Was copied fully (no missing characters)
    - Was saved correctly

If the secret is wrong, Stripe **will deliver the event**, but your store will reject it.

---

5. Check HTTPS & Reachability

Stripe requires:

- Public HTTPS endpoint
- Valid SSL certificate
- No authentication on the webhook URL

Check:

- Visiting `https://YOURDOMAIN/shop/webhook/` returns a response (not a timeout)
- Firewall is not blocking inbound HTTPS
- No IP allow-listing blocking Stripe

---

6. Check Application Logs (Self-Hosted)

Look for:

- Webhook signature verification errors
- 403 / 400 responses
- JSON parsing errors

If Stripe shows delivery but the app errors, logs will show why.

---

7. Retry Delivery from Stripe

In Stripe:

- Open the failed webhook event
- Click **Retry**

If it succeeds on retry:

- The issue was temporary (deploy, restart, DNS)

If it fails again:

- Configuration is still wrong

Confirmation

As long as the above instructions are followed correctly the webhook will always fire.

This really is for information only just in case.

correctly:

- The webhook **will fire**
- Stripe **will deliver**
- Orders **will be created**

Webhook failures almost always come from:

- Wrong URL
- Wrong secret
- Wrong mode
- App not restarted

Not from Stripe itself.

Djangify eCommerce Build

https://djangify.com