

Stripe Webhook Events Configuration Guide

Required Webhook Events for Genesis Provenance

This guide details the exact Stripe webhook events you need to configure in your Stripe Dashboard for Genesis Provenance to function correctly.

Complete Event List

Your Stripe webhook **must** be configured to listen for these **6 events**:

1. `checkout.session.completed`

Purpose: Triggered when a user completes the Stripe Checkout

What it does:

- Creates or updates subscription in database
- Links Stripe customer to organization
- Activates subscription features

Critical for: Initial subscription creation, plan upgrades

2. `customer.subscription.created`

Purpose: Triggered when a new subscription is created

What it does:

- Creates subscription record in database
- Sets initial billing period
- Activates plan features

Critical for: New subscription setup

3. `customer.subscription.updated`

Purpose: Triggered when subscription details change

What it does:

- Updates subscription status (active, past_due, canceled)
- Updates billing period dates
- Syncs plan changes
- Updates payment method status

Critical for: Plan changes, renewals, payment updates

4. `customer.subscription.deleted`

Purpose: Triggered when a subscription is canceled

What it does:

- Marks subscription as canceled in database
- Sets `canceledAt` timestamp
- Triggers feature access revocation

Critical for: Subscription cancellations

5. `invoice.payment_succeeded`

Purpose: Triggered when a subscription payment succeeds

What it does:

- Logs successful payment
- Confirms subscription renewal
- Updates billing period

Critical for: Subscription renewals, payment tracking

6. `invoice.payment_failed`

Purpose: Triggered when a subscription payment fails

What it does:

- Updates subscription status to `past_due`
- Triggers payment retry logic
- Can trigger user notifications

Critical for: Payment failure handling, dunning management



Step-by-Step Webhook Configuration

Step 1: Access Stripe Dashboard

1. Go to <https://dashboard.stripe.com>
2. Log in to your Stripe account
3. Make sure you're in **Test Mode** (toggle in top-right) for testing

Step 2: Navigate to Webhooks

1. Click **“Developers”** in the left sidebar
2. Click **“Webhooks”**
3. You should see your existing webhook endpoint (if any)

Step 3: Find Your Webhook Endpoint

Look for your webhook with the URL:

`https://app.genesisprovenance.com/api/billing/webhook`

If you see it:

- Click on it to edit
- Proceed to Step 4

If you DON'T see it:

- Click “**Add endpoint**” button
- Enter: <https://app.genesisprovenance.com/api/billing/webhook>
- Proceed to Step 4

Step 4: Configure Events to Listen

1. Scroll to “**Events to send**” section
2. Click “**Select events**” or “**Edit**” button
3. In the search box, search for and select each event:

Search and Check These:

- checkout.session.completed
- customer.subscription.created
- customer.subscription.updated
- customer.subscription.deleted
- invoice.payment_succeeded
- invoice.payment_failed

Visual Steps:

- Type `checkout.session.completed` in search
- Check the checkbox
- Repeat for all 6 events

Step 5: Save Configuration

1. Click “**Add events**” or “**Update events**” button at bottom
2. Review your selected events (should show 6)
3. Click “**Add endpoint**” or “**Update endpoint**”

Step 6: Verify Webhook Secret

1. After saving, you'll see your webhook listed
2. Click on the webhook endpoint
3. Look for “**Signing secret**”
4. Click “**Reveal**” to see the secret
5. It should start with `whsec_...`

Verify it matches your `.env` file:

```
STRIPE_WEBHOOK_SECRET=whsec_your_secret_here
```

⚠ If it doesn't match, update your `.env` file!

Verification Checklist

After configuration, verify:

- [] Webhook endpoint URL is: `https://app.genesisprovenance.com/api/billing/webhook`
 - [] All 6 events are selected (see list above)
 - [] `STRIPE_WEBHOOK_SECRET` in `.env` matches Stripe Dashboard
 - [] Webhook is in the correct mode (Test/Live)
 - [] Endpoint status shows as “Enabled”
-

Testing Your Webhook Configuration

Test in Stripe Dashboard

1. Navigate to your webhook

- Developers → Webhooks → Click your endpoint

2. Send Test Webhook

- Click “**Send test webhook**” button
- Select event: `customer.subscription.created`
- Click “**Send test webhook**”

3. Check Response

- You should see **200 OK** response
- If you see errors, check your server logs

Test with Real Subscription

1. Create Test Subscription

- Go to your app: `https://app.genesisprovenance.com/settings/billing`
- Click “Upgrade Plan”
- Select “Dealer” plan
- Use Stripe test card: `4242 4242 4242 4242`
- Complete checkout

2. Verify in Stripe Dashboard

- Developers → Webhooks → Click your endpoint
- Click “**Logs**” tab
- You should see recent webhook events (1-2 minutes ago)
- Status should be **200 OK** (green checkmark)

3. Verify in Your App

- Go to `/settings/billing`
- Your plan should show as “Dealer”
- Status should be “Active”



Troubleshooting

Issue: Webhook returns 401 Unauthorized

Cause: Webhook secret mismatch

Solution:

1. Go to Stripe Dashboard → Developers → Webhooks
2. Click your webhook endpoint
3. Reveal the “Signing secret”
4. Copy it (starts with `whsec_...`)
5. Update your `.env` file:

```
bash
STRIPE_WEBHOOK_SECRET=whsec_your_actual_secret_here
```

6. Restart your application or redeploy
-

Issue: Webhook returns 500 Internal Server Error

Cause: Application error processing webhook

Solution:

1. Check your server logs for detailed error messages
2. Common causes:
 - Database connection issues
 - Missing `organizationId` in subscription metadata
 - Invalid Price ID
3. Test the webhook handler locally:

```
bash
# Check logs
tail -f /var/log/nextjs/application.log | grep webhook
```

Issue: Subscription not updating in app after checkout

Cause: Webhook events not being received

Solution:

1. Verify webhook endpoint URL is correct
 2. Check that all 6 events are selected
 3. Verify webhook is “Enabled” (not disabled)
 4. Check Stripe Dashboard → Webhooks → Logs for failed deliveries
 5. If you see failed attempts, click them to see the error details
-

Issue: Webhook logs show “No matching events”

Cause: Events not configured in webhook settings

Solution:

1. Go to your webhook in Stripe Dashboard
2. Click “Edit” or click the webhook name

3. Scroll to “Events to send”
 4. Click “Select events” or “Edit events”
 5. Add all 6 events from the list above
 6. Click “Update endpoint”
-

Security Best Practices

1. Always Verify Webhook Signatures

Your webhook handler (`/app/api/billing/webhook/route.ts`) automatically verifies signatures using:

```
stripe.webhooks.constructEvent(body, signature, webhookSecret)
```

Never skip this verification!

2. Keep Webhook Secret Secure

- Never commit `.env` file to Git  (already in `.gitignore`)
- Store secret in environment variables only
- Rotate webhook secret if compromised

3. Use Different Webhooks for Test/Live Mode

- **Test Mode:** `whsec_test_...`
- **Live Mode:** `whsec_live_...`

Ensure your production `.env` uses the Live mode secret.

Monitoring Webhook Health

Daily Checks (Recommended)

1. Go to Stripe Dashboard → Developers → Webhooks
2. Click your endpoint
3. Check “**Logs**” tab
4. Verify recent events show **200 OK** (green checkmarks)
5. Look for any failed deliveries (red X)

Set Up Alerts (Optional)

1. In Stripe Dashboard, go to Settings → Notifications
 2. Enable “Webhook endpoint failures”
 3. Enter your email for alerts
-

Production vs. Test Mode

Test Mode

- **Webhook URL:** `https://app.genesisprovenance.com/api/billing/webhook`

- **Webhook Secret:** `whsec_test_...` (from Test Mode dashboard)
- **Use for:** Development, testing, staging

Live Mode

- **Webhook URL:** Same as test (`https://app.genesisprovenance.com/api/billing/webhook`)
- **Webhook Secret:** `whsec_live_...` (from Live Mode dashboard)
- **Use for:** Production, real customer payments

Important: You need to configure webhooks separately in both Test and Live modes!



Quick Reference: Events Summary

Event	Triggers When	Database Action
<code>checkout.session.completed</code>	User completes checkout	Create/update subscription
<code>customer.subscription.created</code>	New subscription starts	Create subscription record
<code>customer.subscription.updated</code>	Subscription changes	Update status, dates, plan
<code>customer.subscription.deleted</code>	Subscription canceled	Mark as canceled
<code>invoice.payment_succeeded</code>	Payment succeeds	Log payment, update period
<code>invoice.payment_failed</code>	Payment fails	Mark as past_due



Final Configuration Checklist

Before going to production:

Test Mode (for development):

- [] Webhook endpoint created for Test Mode
- [] All 6 events selected
- [] Test mode webhook secret in `.env`
- [] Tested with Stripe test cards
- [] Verified 200 OK responses in logs

Live Mode (for production):

- [] Webhook endpoint created for Live Mode
- [] All 6 events selected
- [] Live mode webhook secret in production `.env`
- [] DNS/SSL configured correctly
- [] Tested with real subscription (in test mode first!)
- [] Monitoring/alerts enabled

SOS Need Help?

Stripe Documentation:

- Webhooks: <https://stripe.com/docs/webhooks>
- Testing Webhooks: <https://stripe.com/docs/webhooks/test>
- Event Types: <https://stripe.com/docs/api/events/types>

Genesis Provenance Docs:

- See `CHECKOUT_TROUBLESHOOTING.md` for payment issues
- See `STRIPE_WEBHOOK_SETUP_COMPLETE.md` for webhook secret setup
- See `UPGRADE_CHECKOUT_FIX.md` for subscription sync issues

Stripe Support:

- Dashboard: <https://dashboard.stripe.com/support>
- Email: support@stripe.com

Last Updated: December 2, 2025

App Version: Phase 5C

Webhook Handler: `/app/api/billing/webhook/route.ts`