

# Postally Shopify Integration Guide

---

Version 1.3 | May 1, 2026 | <https://postally.ca>

# 1. Overview

---

This guide explains how to integrate Shopify with Postally to automatically send physical mail in response to Shopify events. The most common use case is abandoned cart recovery — sending a postcard or letter to shoppers who leave items in their cart.

Postally integrates with Shopify in two ways:

- Direct webhook integration — Shopify sends events directly to a Postally trigger URL (recommended for developers)
- Zapier / Make.com — No-code integration using Shopify triggers and Postally actions (recommended for non-technical users)

## 2. How It Works

---

Postally's inbound trigger system receives HTTP POST requests from external services and automatically creates and sends mail. Here is the flow:

1. A Shopify event occurs (e.g., a checkout is created, an order is placed).
2. Shopify sends a webhook POST request to your Postally trigger URL.
3. Postally extracts the recipient address from the webhook payload using your field mapping.
4. Postally creates a mailpiece using your configured template and sends it to print.
5. The postcard or letter is printed, posted, and delivered to the customer.

The entire process is automatic after initial setup. No manual intervention required.

## 3. Abandoned Cart Recovery — Step by Step

---

Follow these steps to set up automated abandoned cart postcards or letters.

### 3.1 Create a Template

1. Log in to the Postally dashboard at <https://postally.ca>.
2. Navigate to Templates and click Create Template.
3. Choose type: postcard (recommended for cart recovery) and select a format (e.g., postcard\_6x9 for maximum visual impact).
4. Design your template with a compelling message and a clear call-to-action. Use merge variables for personalization:

```
Hey {{first_name}}, you left something in your cart!
```

```
Use code COMEBACK10 for 10% off your order.
```

```
Visit {{cart_url}} to complete your purchase.
```

5. Click Save, then Publish to make the template available.

**Note:** You can include a QR code that links back to their cart for easy mobile checkout.

### 3.2 Create an Automation (Dashboard)

The easiest way to set up the integration is through the Automations page in the dashboard:

1. Navigate to Automations in the sidebar.
2. Click "New Automation".
3. Select the "Abandoned Cart" category.
4. Give your automation a name (e.g., Abandoned Cart Follow-up).
5. Select the template you created in step 3.1.
6. Choose the mailpiece type and format.
7. Configure the field mapping to tell Postally where to find address data in the Shopify webhook payload (see section 4 for the mapping reference).
8. Click "Create". You will receive a trigger URL and a secret.

**Important:** Save the secret immediately — it is shown only once. You will need it to configure Shopify.

### 3.3 Create an Automation (API)

Alternatively, create the trigger via the API:

```
POST https://api.postally.ca/v1/triggers/create
```

```
-H "Authorization: Bearer dm_live_your_key"
```

```
-H "Content-Type: application/json"
```

```
-d '{
```

```
  "name": "Shopify Abandoned Cart",
```

```
  "template_id": "tmpl_your_template_id",
```

```
  "mailpiece_type": "postcard",
```

```
  "format": "postcard_6x9",
```

```
  "field_mapping": {
```

```
    "first_name": "shipping_address.first_name",
```

```
    "last_name": "shipping_address.last_name",
```

```
    "address_line1": "shipping_address.address1",
```

```
    "address_line2": "shipping_address.address2",
```

```
    "city": "shipping_address.city",
```

```
    "state": "shipping_address.province_code",
```

```
    "postal_code": "shipping_address.zip",
```

```
    "country": "shipping_address.country_code"
```

```
  }
```

```
}'
```

The response includes:

- `inbound_url` — The URL Shopify will POST to (e.g., `https://api.postally.ca/v1/triggers/inbound/<id>`)

- secret — A unique secret for authenticating requests (shown only once)

### 3.4 Configure Shopify Webhooks

1. Log in to your Shopify Admin at <https://admin.shopify.com>.
2. Go to Settings > Notifications > Webhooks.
3. Click "Create webhook".
4. Select the event: "Checkout creation" (fires when a customer starts checkout but has not completed it).
5. Set the format to JSON.
6. Paste your Postally trigger URL from step 3.2 or 3.3.
7. Save the webhook.

Authentication: Postally verifies the trigger secret. Include it as the X-Trigger-Secret header:

```
https://api.postally.ca/v1/triggers/inbound/<id>?secret=your_trigger_secret
```

**Tip:** To avoid sending postcards to customers who complete their purchase, add a delay in your workflow using Shopify Flow or Zapier. Check if the checkout was completed before triggering the Postally send.

### 3.5 Test the Integration

1. Use a Postally test API key (dm\_test\_...) when creating the trigger for initial testing.
2. Create a test checkout in Shopify (add items to cart, enter a shipping address, but don't complete payment).
3. Shopify will fire the webhook to your Postally trigger URL.
4. Check the Postally dashboard > Mailpieces to confirm a test mailpiece was created.
5. Verify the recipient address was correctly mapped from the Shopify data.
6. Once verified, recreate the trigger with a live API key for production.

## 4. Shopify Field Mapping Reference

---

When configuring a Postally trigger, you specify a `field_mapping` object that maps Postally recipient fields to JSON paths in the Shopify webhook payload. Use dot notation to access nested fields.

### 4.1 Checkout / Abandoned Cart Webhook

Shopify's checkout webhook payload includes shipping address data. Here are the recommended mappings:

| Postally Field             | Shopify JSON Path                           |
|----------------------------|---|
| <code>first_name</code>    | <code>shipping_address.first_name</code>    |
| <code>last_name</code>     | <code>shipping_address.last_name</code>     |
| <code>company</code>       | <code>shipping_address.company</code>       |
| <code>address_line1</code> | <code>shipping_address.address1</code>      |
| <code>address_line2</code> | <code>shipping_address.address2</code>      |
| <code>city</code>          | <code>shipping_address.city</code>          |
| <code>state</code>         | <code>shipping_address.province_code</code> |
| <code>postal_code</code>   | <code>shipping_address.zip</code>           |
| <code>country</code>       | <code>shipping_address.country_code</code>  |
| <code>email</code>         | <code>email</code>                          |

### 4.2 Order Webhook (Thank-You Cards)

For order-based automations (e.g., thank-you cards after purchase), use the `orders/create` event:

| Postally Field             | Shopify JSON Path                        |
|----------------------------|--|
| <code>first_name</code>    | <code>shipping_address.first_name</code> |
| <code>last_name</code>     | <code>shipping_address.last_name</code>  |
| <code>company</code>       | <code>shipping_address.company</code>    |
| <code>address_line1</code> | <code>shipping_address.address1</code>   |

| Postally Field | Shopify JSON Path              |
|----------------|--------------------------------|
| address_line2  | shipping_address.address2      |
| city           | shipping_address.city          |
| state          | shipping_address.province_code |
| postal_code    | shipping_address.zip           |
| country        | shipping_address.country_code  |

**Note:** Shopify webhook payloads use address1/address2 (not address\_line1/address\_line2). The field mapping translates between the two.

### 4.3 Custom Merge Variables

You can also map Shopify data to template merge variables for personalized content:

| Template Variable | Shopify JSON Path           |
|-------------------|-----------------------------|
| {{first_name}}    | shipping_address.first_name |
| {{order_number}}  | order_number                |
| {{total_price}}   | total_price                 |
| {{cart_url}}      | abandoned_checkout_url      |
| {{shop_name}}     | source_name                 |

## 5. Other Shopify Use Cases

---

### 5.1 Order Thank-You Cards

Send a physical thank-you postcard after every purchase to build customer loyalty.

- Shopify event: orders/create
- Template: Thank-you message with customer name and order number
- Mail type: Postcard (4x6 or 6x9)

### 5.2 Win-Back Campaigns

Re-engage lapsed customers who have not purchased in a set period.

- Use Shopify Flow or a Zapier schedule to identify inactive customers
- Export the list and create a Postally campaign via the API
- Include a personalized discount code to incentivize return

### 5.3 VIP Customer Postcards

Reward high-value customers with exclusive physical mail.

- Use Shopify customer tags or order total thresholds to identify VIPs
- Send branded postcards with loyalty rewards or early access offers

### 5.4 Seasonal and Holiday Mailers

Send seasonal direct mail campaigns to your Shopify customer base.

- Export customers from Shopify and import into Postally as a recipient group
- Use the Campaigns feature to send personalized self-mailers or postcards at scale

## 6. No-Code Integration with Zapier / Make.com

---

If you prefer a no-code setup, use Zapier or Make.com to connect Shopify and Postally.

### 6.1 Zapier Setup

1. Create a new Zap in Zapier.
2. Set the trigger to "Shopify: New Abandoned Cart" (or "New Order" for thank-you cards).
3. Set the action to "Postally: Send Postcard" (letters coming soon).
4. Connect your Postally account using your API key.
5. Map the Shopify address fields to the Postally recipient fields.
6. Select your template and format.
7. Test the Zap and turn it on.

**Tip:** Zapier's built-in delay feature lets you wait (e.g., 24 hours) before sending, so you only mail customers who truly abandoned their cart.

### 6.2 Make.com Setup

The process is similar in Make.com:

1. Create a new scenario with a Shopify trigger module.
2. Add an HTTP module to POST to the Postally API (or use the Postally integration if available).
3. Map address fields and configure the template.
4. Add a delay module if needed for abandoned cart timing.
5. Activate the scenario.

## 7. Troubleshooting

| Issue                  | Solution  |
|------------------------|---|
| Webhook not firing     | Verify the webhook is active in Shopify Admin > Settings > Notifications > Webhooks. Check that the URL is correct.   |
| Mailpiece not created  | Check the Postally dashboard > Automations to confirm the trigger is active. Verify the trigger URL and secret are correct.                                 |
| Address mapping errors | Review the field_mapping object. Shopify uses address1/address2 (not address_line1). Use dot notation for nested fields.                                    |
| Missing address data   | Some Shopify checkouts may not include a shipping address (e.g., digital products). Postally requires at minimum address_line1, city, and postal_code.      |
| Duplicate mailpieces   | Use Postally's Idempotency-Key header when calling the API directly. For trigger-based sends, Postally deduplicates by trigger + recipient within 24 hours. |
| Test mode not working  | Ensure you created the trigger with a test API key (dm_test_...). Test triggers create test mailpieces that are not printed.                                |

### 7.1 Checking Trigger Activity

The Automations page in the Postally dashboard shows each trigger's fire count and last fired timestamp. Use this to confirm webhooks are being received.

### 7.2 Getting Help

Contact [support@postally.ca](mailto:support@postally.ca) for integration assistance. Include your trigger ID and any error messages.