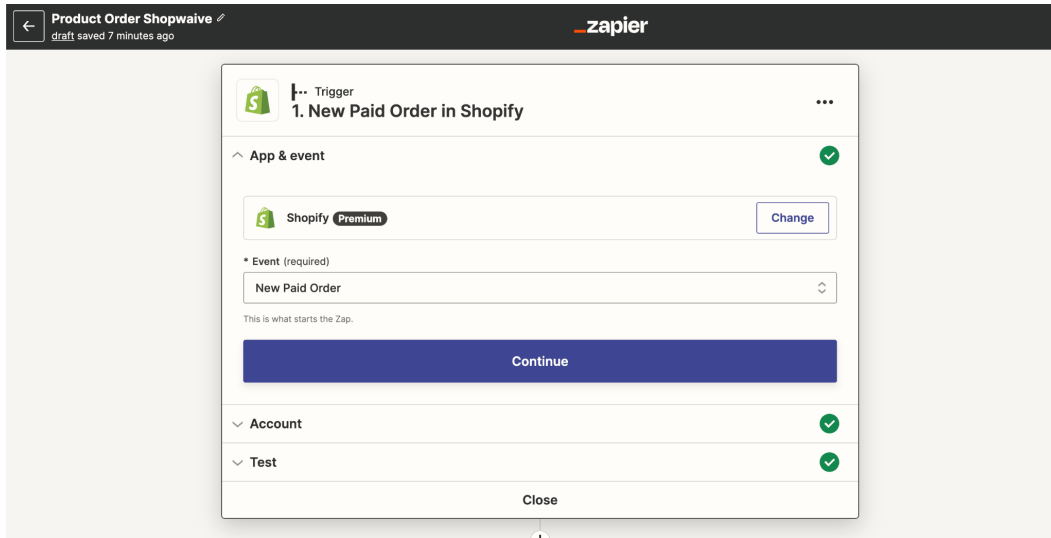
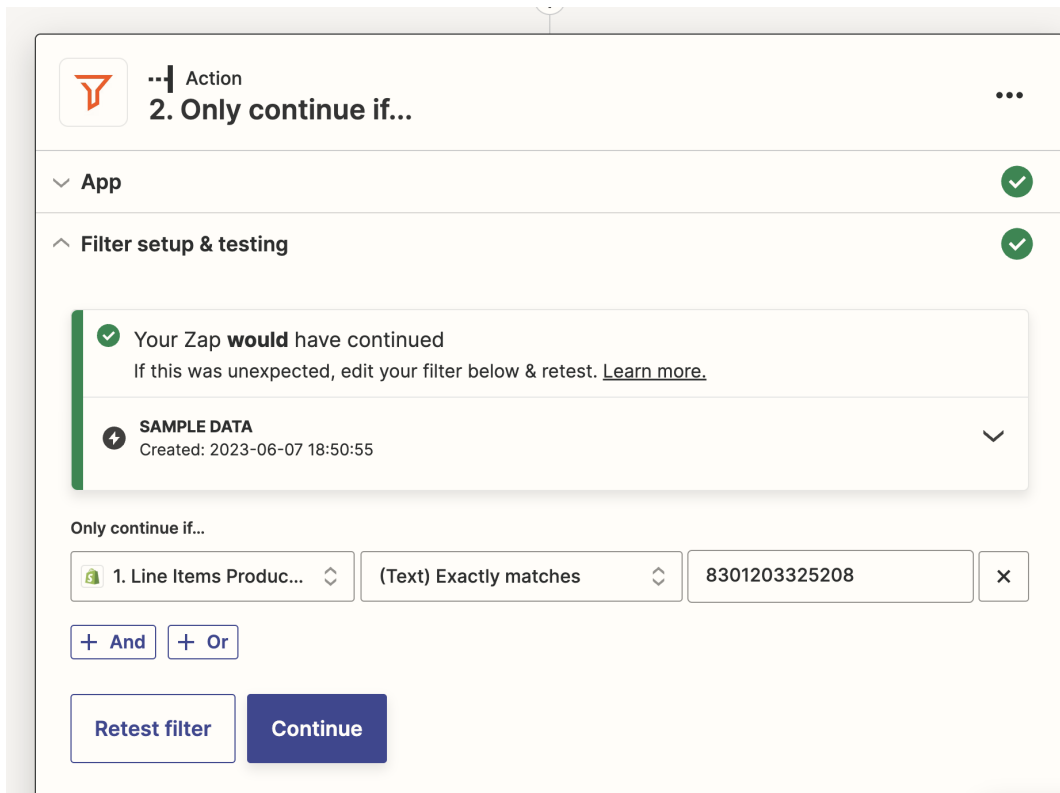




STEP 1: Trigger New Paid Order in Shopify



STEP 2: Filter. Only continue if Line Item Product ID (Text) Exactly matches the ID or variant you'd like to associate with a top up





STEP 3: Create Loop from Line items. Select the Product ID and associated purchased quantity for that ID, the line item price, and the customer's email address who purchased

Product Order Shopwaive  

draft saved 10 minutes ago

3. Create Loop From Line Items in Looping by Zapier










App & event 

Action 


WARNING: Looping by Zapier is an advanced feature currently in Beta. Learn [how to create loops here](#) and [understand limitations here](#).


Line items

* Values to Loop (required)

productid	 1. Line Items Product Id: 8301203325208	
quantity	 1. Line Items Quantity: 3	
price	 1. Line Items Price: 75.00	
email	 1. Email: support@shopwaive.com	
		

Type loop value name(s) in the field(s) on the left. Map Line Item value(s) to loop through on the right.

Trim Whitespace 

True 

Remove whitespace surrounding each individual value? Defaults to 'True'.

Loop iteration counter start ¹ ² ³

1

A counter value called 'loop_iteration' will be added to each iteration of the loop. Does not affect the data to loop... [more](#)

Maximum number of Loop iterations ¹ ² ³

500

Set this value to limit the number of loops performed. Data in iterations past the limit will be ignored. A number higher... [more](#)

STEP 4: Format the output of the last step, by multiplying the Price, Quantity, and any Multiplier you choose. In the example below, we chose “10”, which means Purchase amount for SKU = \$225, then the top up credit equals $\$225 \times 10 = \2250 . So if the customer originally had a \$100 balance before the product purchase, then the new balance after purchase would be \$2350.

The screenshot shows the Zapier interface for configuring an action. At the top, the header includes a back arrow, the text "Product Order Shopwaive" with a pencil icon, and "draft saved 10 minutes ago". The Zapier logo is on the right. Below the header is a plus sign. The main content area is titled "Action 4. Numbers in Formatter by Zapier". It has a "App & event" section with a dropdown arrow and a green checkmark. Below that is an "Action" section with a dropdown arrow and a green checkmark. The action is "Perform Math Operation". A tooltip provides instructions: "Perform mathematical operations on value(s). If you have [line items](#), use [Spreadsheet-Style Formula](#) instead." The "Operation" is set to "Multiply". Below this is a section for "Values" with the label "The math operation to perform." Under "Values", there is an "Input" section with three fields: "3. Price: 75.00", "3. Quantity: 3", and "10". Each field has a small 'x' icon to its right. Below these fields is a text input field with the placeholder "Enter text or insert data...". At the bottom of the "Values" section is the text "Numbers you would like to use in the math operation." There are two buttons: "Refresh fields" and "Continue". At the bottom of the interface, there is a "Test" section with a dropdown arrow and a green checkmark, and a "Close" button.

STEP 5: Complete a Webhook POST to the Shopwaive API at the url listed below. Include the data fields labeled as shown. You may choose to include any note text in the note field as you like. In the “store” field, you need to input your store abbreviated Shopify url. For instance, set store equal to **cmpfire** if your shopify url is “cmpfire.myshopify.com”

In the Header, include the **X-Shopwaive-Access-Token** header, with value set equal to your Shopify store Access Token header, X-Shopify-Access-Token.

The screenshot shows the Zapier configuration page for a 'Product Order Shopwaive' draft. The main section is titled '5. POST in Webhooks by Zapier'. Under 'App & event', 'Webhooks by Zapier' is selected. Under 'Action', 'POST' is selected. The 'URL' field is set to 'https://app.shopwaive.com/api/purchase/product/topup'. The 'Payload Type' is set to 'json'. The 'Data' section contains the following key-value pairs: 'amount' with value '4. Output: 2250', 'note' with value 'Topup for 3. Productid: 8301203325208', 'email' with value '1. Email: support@shopwaive.com', and 'store' with a redacted value. The 'Wrap Request in Array' option is set to 'No'. The 'File' field is empty. The 'Unflatten' option is set to 'Yes'. The 'Headers' section contains one header: 'X-Shopwaive-Access-' with a redacted value. At the bottom, there are 'Refresh fields' and 'Continue' buttons.

Product Order Shopwaive
draft saved 9 minutes ago

zapier

5. POST in Webhooks by Zapier

App & event

Action

* URL (required)
https://app.shopwaive.com/api/purchase/product/topup
Any URL with a querystring will be re-encoded properly.

Payload Type
json

Pay special attention to the proper mapping of the data below.

Data

amount	4. Output: 2250	x
note	Topup for 3. Productid: 8301203325208	x
email	1. Email: support@shopwaive.com	x
store	[REDACTED]	x

These are key-value pairs that will form the data portion of the request. If you leave these empty, all fields... more

Wrap Request in Array
No

Wraps the request in an array if that is something the service requires.

File
Enter text or insert data...
A file object to be attached to the request. The request will automatically be converted to multipart/form... more

Unflatten
Yes

By default we convert fields with double underscores into nested dictionaries. Select no here to turn this... more

Headers

X-Shopwaive-Access-	[REDACTED]	x
---------------------	------------	---

Key, value pairs to be added as headers in all requests.

Refresh fields Continue