

# Shopify Event Triggers

Agile CRM integrates with a wide range of plugins, including Shopify. Our Shopify integration helps you supercharge your Shopify store with smart customer management, powerful campaigns, personalized web engagement & more. You can set up sync between Shopify & the CRM to sync your Shopify contacts data into Agile on a daily, weekly, monthly or one time basis.

We now have introduced “**Triggers**” that can be set off whenever a Shopify Event occurs, to automate specific campaigns on your customers. (sending an email, setting up a task/ follow up call etc···)

List of available Shopify Events are as follows:

- Customer created -
- Customer updated -
- Order created -
- Order updated -
- Order payment -
- Order deleted -
- Order fulfilled -
- Order cancelled -
- Checkout created -
- Checkout updated -
- Checkout deleted -

To set up a Shopify Event Trigger in Agile, please follow the steps given below:

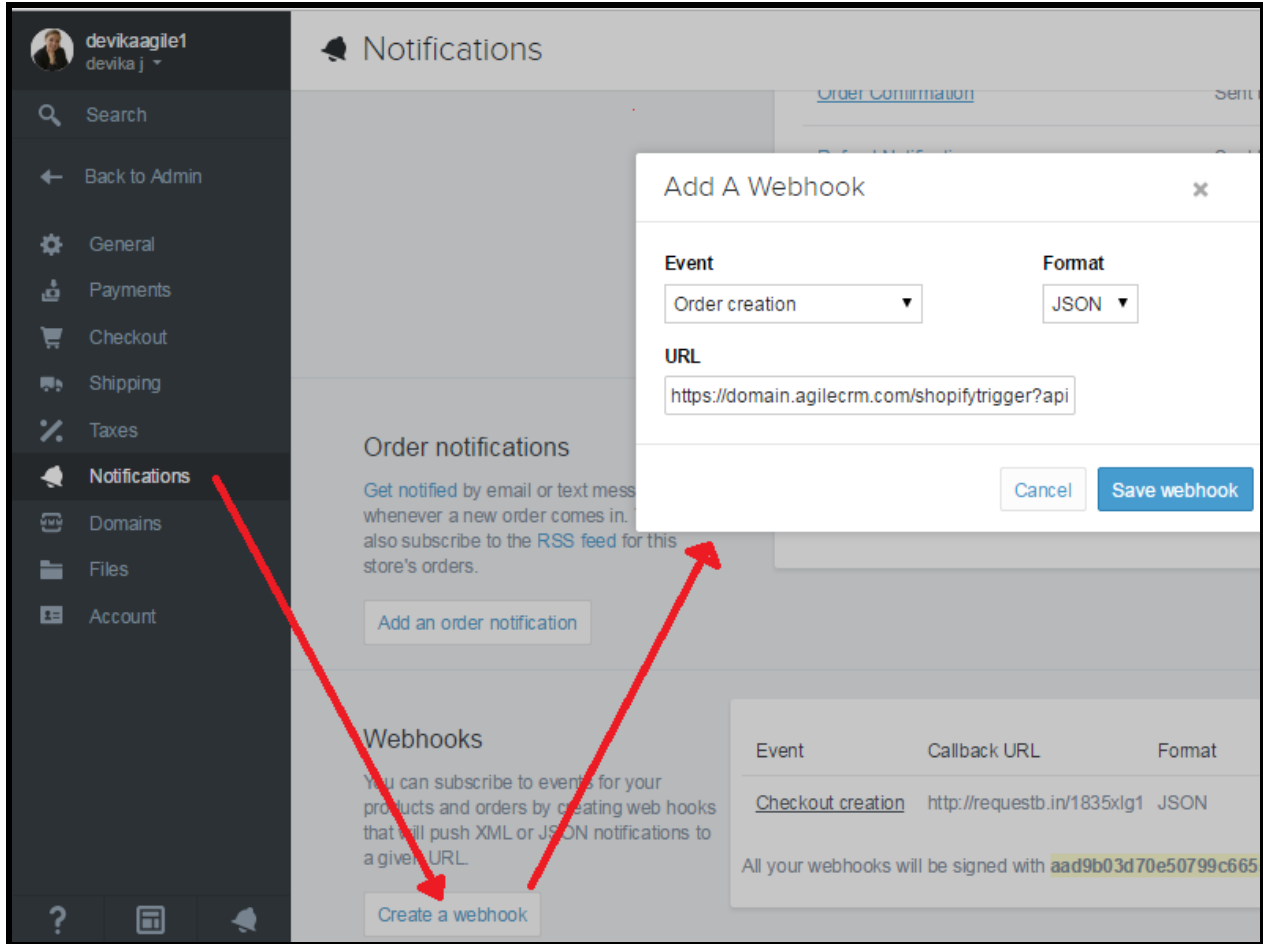
## 1. Create Webhooks in Shopify

Firstly, you have to create a webhook in Shopify for each Shopify event.

- a. To do that, go to your Shopify dashboard and click on the ‘Notifications’ in the menu on the left side ---> ‘Create a Webhook’ button.
- b. Now, in the popup window, choose your required Shopify Event, set the format to ‘JSON’ and enter the URL as follows:

URL: <https://domain.agilecrm.com/shopifytrigger?api-key=API-Key>

(‘domain’ and ‘API-Key’ in the above URL have to be replaced with your domain and Agile API-Key respectively.)



### Add A Webhook ✕

**Event**  
Order creation ▼

**Format**  
JSON ▼

**URL**

Cancel Save webhook

c. You have to do this for all the Shopify events that you need.

## 2. Setting up Triggers in Agile

- a. After setting up webhooks in your Shopify account, from your Agile dashboard, go to Campaigns ---> Manage Triggers ---> Add Trigger.
- b. From the '+Add Trigger' window, select 'Shopify Event' in 'When this happens' field and then select your Shopify Event.
- c. Next, select the campaign that you want to run on the select audience and save the details.

**+ Add Trigger**

Name \*

When this happens \*

- Order updated
- Select Shopify Event
- Customer created
- Customer updated
- Order created
- Order updated**
- Order payment
- Order deleted
- Order fulfilled
- Order cancelled
- Checkout created
- Checkout updated
- Checkout deleted

Run this Campaign on the Contact \*

Save

Shopify account for this. In your stores Admin... hooks, and create a webhook for the desired event

?api-key=3vlbhlqmb24m69dltcej0vboa2

You're now all set up with Shopify event triggers.

Setting up Shopify triggers will fully avoid the need for syncing your Shopify contacts to Agile, after the initial sync if you haven't set up the regular sync.

## Usage

### Available Merge fields

Whenever you run Campaigns using the Shopify Event Triggers, you can use data provided by shopify in your campaign emails or conditions. For example, you can include a merge field like `{{shopify.abandoned_checkout_url}}` in your campaign mail to include a link to the abandoned cart and get your customers complete their order.

Various type of merge fields are available depending on the Trigger Event configured. Here's the list:

Shopify event	Merge Fields
Customer creation	"accepts_marketing", "created_at", "email", "first_name", "id", "last_name", "last_order_id", "multipass_identifier", "note", "orders_count", "state", "total_spent", "updated_at", "verified_email", "tags", "last_order_name", "addresses"  Shopify <a href="#">customer object</a> is available.
Customer update	Shopify <a href="#">customer object</a> is available.
Order created	Shopify <a href="#">order object</a> is available.
Order updated	Shopify <a href="#">order object</a> is available.
Order payment	Shopify <a href="#">order object</a> is available.
Order deletion	"id", "shipping_lines", "line_items", "billing_address", "shipping_address", "fulfillments", "refunds", "customer"
Order fulfillment	Shopify <a href="#">order object</a> is available.
Order cancelled	Shopify <a href="#">order object</a> is available.
Checkout created	Shopify <a href="#">order object</a> is available.
Checkout updated	
Checkout Deleted	

To use merge fields in your campaign mails, they have to be written as:  
`{{shopify.event.merge_field}}`

A few examples of the merge field representation: `{{shopify.customer.first_name}}`;  
`{{shopify.customer.email}}`; `{{shopify.oder.quantity}}`; `{{shopify.order.variant_id}}`;  
`{{shopify.billing_address.company}}` & so forth.

## Use Cases

### How to setup Cart Abandonment campaign?

Whenever your customer adds a few items to the cart, but, abandons the cart before making the purchase, you can run an automated campaign to remind him of the purchase, also given him an incentive to make the purchase if required.

Here's how you set up the campaign:

Firstly, create a simple campaign workflow as follows:

Insert "Add/ Remove Tag' node that adds some tag (eg: product selected) & link it to the start node. Following that, insert a 'Wait Node' with a duration of 1 day. Now, insert a 'Check Tag' node that checks the previous tag (here, 'product selected') after that and add a 'Send Email' node to the 'Yes' terminal of the 'Check Tag' node and save the campaign. (See the workflow below.)



In the the 'Send Email' node, write down your required text and include the link to the abandoned cart with the merge field - `{{shopify.abandoned_checkout_url}}` and fill in the other required details. Use the merge field - `{{email}}` in the 'Send to' field.

Configure the webhook for 'Checkout Created' event, and set up the Shopify Event trigger for the same event. Select the campaign that you just created in the Event trigger & save it.

Create another campaign with 'Add/ Remove Tag' node that removes the same tag mentioned before, (eg: product selected - the same as the above one) from the contact. Now, configure another webhook for 'Checkout Updated' event, and set up the Shopify Event Trigger for the same event. Select the campaign that you just created in the Event trigger & save it.



That's it. You are set with Cart Abandonment!

Logic: When the 'Checkout Created' event happens, the first campaign adds the tag - 'product selected' to the contact & waits for a day. During this wait period of 1 day, if the contact has made the payment, the 'Checkout Updated' event happens. This event triggers the second campaign which removes the previously added tag ('product selected' here) and the mail in the first campaign is not sent as he has made the purchase.

Incase, he has really abandoned the cart (fails to make purchase within the wait period), the second campaign is not set off and the mail in the first campaign gets sent.

Thanks for checking out!