

# Shopify Get Products Activity

## Introduction

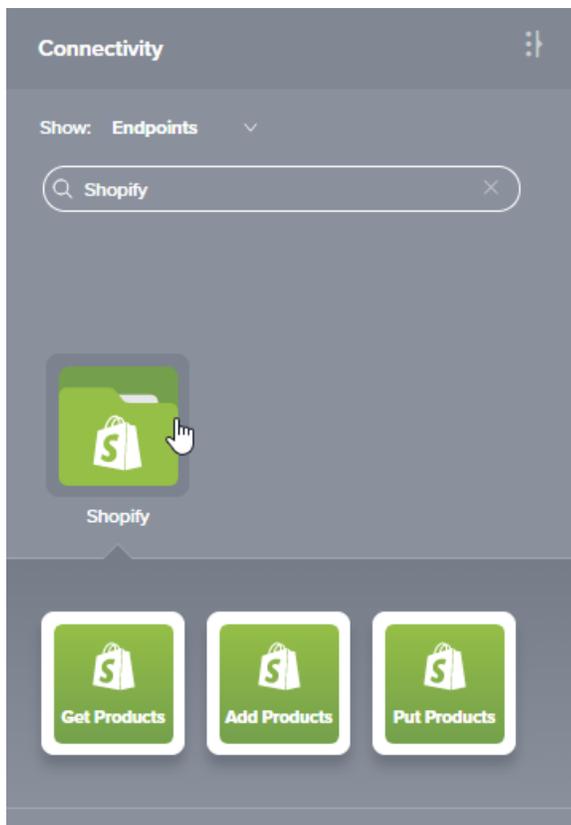
A Shopify get products activity retrieves all product data from a Shopify connection and is used as a source in an operation. After configuring a Shopify connection, you can configure as many Shopify activities as you like for each Shopify connection.

## Creating a Shopify Activity

From the design canvas, open the **Connectivity** tab of the design component palette:



Within the **Endpoints** filter, click the Shopify connection block to display activities that are available to be used with a Shopify connection:



To create an activity that can be configured, the activity must first be added to an operation on the design canvas. To add an activity to an operation, drag the activity block from the palette to the operation.

For more information about the parts of an operation and adding activities to operations, see [Operation Creation and Configuration](#).

## Accessing Menu Actions

After a Shopify activity has been added to an operation, menu actions for that activity are accessible from the project pane in either the **Workflows** or the **Components** tabs, and from the design canvas:

- **Project Pane:** In the **Workflows** or **Components** tab of the project pane, hover over an activity name and click the actions menu icon to open the actions menu.
- **Design Canvas:** Within the operation, click an existing activity block to open the actions menu.

Each of these menu actions is available:

### On This Page

- [Introduction](#)
- [Creating a Shopify Activity](#)
- [Accessing Menu Actions](#)
- [Configuring a Shopify Get Products Activity](#)
  - [Step 1 – Enter Basic Information](#)
  - [Step 2 – Review Data Schema](#)
- [Next Steps](#)

### Related Articles

- [Component Dependencies, Deletion, and Removal](#)
- [Component Deployment](#)
- [Component Palette](#)
- [Design Canvas](#)
- [Operation Creation and Configuration](#)
- [Operation Options](#)
- [Operation Validity](#)
- [Plugins Added to an Activity](#)
- [Project Pane](#)
- [Shopify Add Products Activity](#)
- [Shopify Connection](#)
- [Shopify Put Products Activity](#)
- [Shopify Registration](#)
- [Transformation Creation and Configuration](#)

### Related Topics

- [Cloud Studio](#)
- [Connectors](#)
- [Operations](#)
- [Scripts](#)
- [Shopify](#)
- [Variables](#)

Last updated: Jun 25, 2019

- **View/Edit:** This opens the activity configuration screen for you to configure the activity. For details, see *Configuring a Shopify Get Products Activity* later on this page.
- **Delete:** This is used to permanently delete the activity.
- **Rename:** This positions the cursor on the activity name in the project pane for you to make edits.
- **View Dependencies:** This changes the view in the project pane to display any other parts of the project that the activity is dependent on.
- **Remove:** Available only from the actions menu on the design canvas, this removes the activity as a step in the operation without deleting it from the project. When you remove an activity that is adjacent to a transformation, if schemas are provided within that activity, they will no longer be referenced by the transformation. Removed components can be accessed or permanently deleted from the project pane.
- **Deploy:** This deploys the activity and any components it is dependent on.
- **Configurable Deploy:** This opens the deployment screen, where you can select project components to deploy.
- **Add to group:** This opens a prompt to create a new custom group or to add the component to an existing group.
- **Duplicate:** This creates a copy of the activity as a new, unreferenced component. Upon creating the component copy, the cursor is positioned on the component name within the project pane for you to rename the component.

## Configuring a Shopify Get Products Activity

Follow these steps to configure a Shopify get products activity:

- Step 1 – Enter Basic Information
- Step 2 – Review Data Schema

### Step 1 – Enter Basic Information

- **Name:** Enter a name to use to identify the Shopify get products activity. The name must be unique for each get products activity and must not contain forward slashes (/) or colons (:).
- **Save & Exit:** If enabled, click to save the configuration for this step and close the activity configuration.
- **Next:** Click to temporarily store the configuration for this step and continue to the next step. The configuration will not be saved until you click the **Finished** button on the last step.
- **Discard Changes:** After making changes, click **Discard Changes** to close the configuration without saving changes made to any step. A message will ask you to confirm that you want to discard changes.

### Step 2 – Review Data Schema

- **Data Schema:** The response data schema for Shopify will be displayed. If the operation uses a transformation, the data schemas will be displayed again later during the transformation mapping process, where you can map to target fields using source objects, scripts, variables, custom values, and more.

The Shopify connector uses the [Shopify REST Admin Product API](#). Refer to the API documentation and links in the following table for information on the schema fields.

- **Response:**

Response Schema Field /Node	API Field or Type	Notes
GetAllProductsResponse	node	Response from the activity
product	node	Node representing a <a href="#">product</a>
id	Integer	A unique numeric identifier, of up to 32 digits, for the product
title	String	The name of the product
product_type	String	A categorization for the product used for filtering and searching products
vendor	String	The name of the product's vendor
published_scope	String	The date and time ( <a href="#">ISO 8601 format</a> ) when the product was published
updated_at	String	The date and time ( <a href="#">ISO 8601 format</a> ) when the product was last modified
variants	node	A list of <a href="#">product variants</a> , each representing a different version of the product
price	String	The price of the product variant
inventory_policy	String	Whether customers are allowed to place an order for the product variant when it's out of stock
taxable	Boolean	Whether a tax is charged when the product variant is sold
requires_shipping	Boolean (Deprecated)	Whether a customer needs to provide a shipping address when placing an order for the product variant

- **Add Plugin(s):** Plugins are Jitterbit- or user-provided applications that extend Harmony's native capabilities. To apply a plugin to the activity, click to expand this section and select the checkbox next to the plugin to be used. For additional instructions on using plugins, including details on setting any required variables used by the plugin, see [Plugins Added to an Activity](#).
- **Back:** Click to temporarily store the configuration for this step and return to the previous step.
- **Finished:** Click to save the configuration for all steps and close the activity configuration.
- **Discard Changes:** After making changes, click **Discard Changes** to close the configuration without saving changes made to any step. A message will ask you to confirm that you want to discard changes.

## Next Steps

After configuring a Shopify get products activity, complete the configuration of the operation by adding and configuring other activities, transformations, or scripts as operation steps. You can also configure an operation's operation settings, which include the ability to chain operations together that are in the same or different workflows.

Shopify get products activities can be used as a source with these operation patterns:

- [Transformation Pattern](#)
- [Two-Target Archive Pattern](#) (as the first source only)
- [Two-Transformation Pattern](#) (as the first or second source)

Other patterns are not valid using Shopify get products activities. See the validation patterns on the [Operation Validity](#) page.

To use the activity with scripting functions, write the data to a temporary location and then use that temporary location in the scripting function.

When ready, deploy and run the operation and validate behavior by checking the operation logs.